

Manual Tecnico

Yefer Rodrigo Miguel Alvarado Tzul

Carné: 201731163

Universidad San Carlos de Guatemala

Centro Universitario de Occidente (CUNOC)

División de Ciencias de la Ingeniería

Organizaciones de Lenguajes y Compiladores 2

Ing. Moises Granados

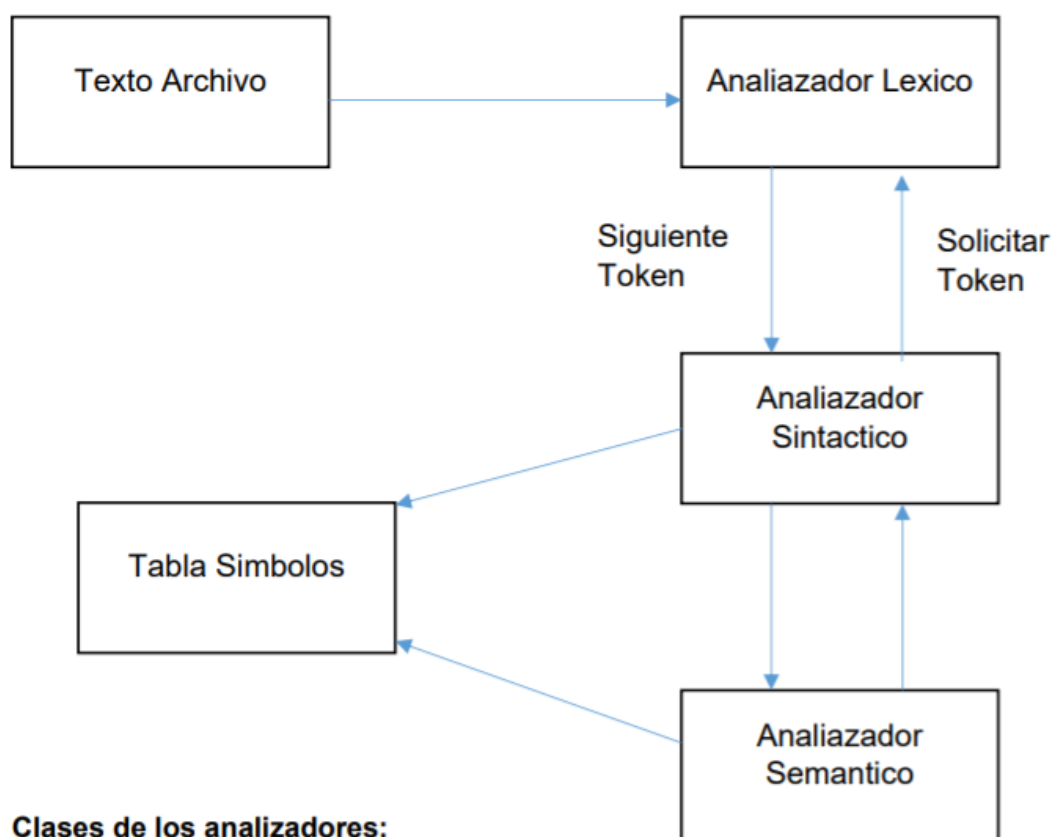
10 de septiembre de 2021

Manual Tecnico

DESCRIPCIÓN DEL PROGRAMA

El programa cuenta con un analizador léxico y sintáctico. Estos se usan para determinar los objetos que se crearán para integración del componente del programa. Cuenta con dos analizadores sintácticos ya que, uno se encarga de los errores y otro de recolectar los datos para el ast. Y un analizador léxico para darle color a las palabras.

Tenemos un archivo de lectura el cual nos encargamos de guardar luego.



Clases de los analizadores:

AnalizadorLexicoCode

AnalizadorSintacticoCode

AnalizadorLexico

AnalizadorSintactico

AnalizadorLexicoTS

AnalizadorSintacticoTS

Implementación del IDE

El ide cuenta con una ventana, en esta se agrega el código fuente, para posteriormente este sea analizada, y en caso de que existan errores abrirá una ventana de errores.

GRAMÁTICA

Analizador Léxico

Terminales

PISTA, IDENTIFICADOR, EXTIENDE, COMA, PUNTO_COMA, ENTERO, DOBLE, BOOLEAN, CHARACTER_RSV, CADENA_RSV, IGUAL_IGUAL, DIFERENTE, MAYOR_Q, MENOR_Q, MAYOR_IGUAL, MENOR_IGUAL, IS_NULL, AND, NAND, OR, NOR, XOR, NOT, MAS, MENOS, POR, DIVISION, MODULO, POTENCIA, KEEP, VAR, IGUAL, NUMERO, DECIMAL, CADENA, TRUE, FALSE, CHARACTER, MAS_IGUAL, MAS_MAS, MENOS_MENOS, ARREGLO, CORCHETE_A, CORCHETE_C, LLAVE_A, LLAVE_C, SI, SINO, PA_A, PA_C, SWITCH, CASO, SALIR, DEFAULT, PARA, MIENTRAS, HACER, CONTINUAR, RETORNAR, VOID, REPRODUCIR, ESPERAR, ORDENAR, ASCENDENTE, DESCENDENTE, PARES, IMPARES, PRIMOS, SUMARIZAR, LONGITUD, MENSAJE, PRINCIPAL, DO, RE, MI, FA, SOL, LA, DO_S, RE_S, FA_S, SOL_S, LA_S, SINO_SI, INDENT, DEDENT.

Analizador Sintactico

no Terminales

INICIO, blanco, codigo, struct_pista, struct_extiende, extiende_state, struc_code, struct_declaracion_var, boolean keep_state, valor_declaracion_operacion, valor, struc_asig_var, tipo_dato, struct_operador_logico, valor_operacion, struct_declaracion_array, dim_arreglo, valor_arreglo, valores_array, struct_si, struct_sino, struct_switch, state_tab, struct_asig_arreglo, struct_para, struct_mientras, struct_hacer, struct_funcion,

struct_procedimiento, struct_reproducir, struct_esperar, struct_ordenar, struct_sumarizar,
struct_longitud, struct_mensaje, struct_principal, struct_llamada_funcion,
struct_llamada_arreglo, parametros, state_params, para_asig_var, condicion_para,
incremento_para, struct_parametros, notas, tipo_orden, struct_operador_relacional,
tipo_valor_op_logico, mensajes, st_ind, st_ded, st_final, code_funcion, code_metodo,
caso_salir, state_continuar, struct_caso, params_lleno, param_vacio,
valores_array_mas_dim, especiales_q_retornan, state_op_logicos, state_op_racional,
struct_declaracion_var_f, struct_declaracion_array_f.

Gramática Sintáctica.

INICIO ::= blanco codigo;

blanco ::= INDENT blanco DEDENT blanco
|;

ESTADOS DE INDENT y DEDENT

st_ind ::= INDENT;

st_ded ::= DEDENT;

st_final ::= DEDENT
|;

COMENTARIOS

CODIGO PISTA

```
codigo ::= struct_pista codigo  
|;
```

ESTRUCTURA DE PISTA

```
struct_pista ::=  
    PISTA IDENTIFICADOR extiende_state st_ind struc_code st_final;
```

ESTADO EXTIENDE O NO

```
extiende_state ::= struct_extiende  
|;
```

ESTRUCTURA EXTIENDE

```
struct_extiende ::= struct_extiende COMA IDENTIFICADOR  
|EXTIENDE IDENTIFICADOR;
```

```
struc_code ::= struct_declaracion_var struc_code  
|struct_declaracion_array struc_code  
|struct_funcion struc_code  
|struct_procedimiento struc_code
```

|struct_principal struc_code

|;

ESTRUCTURA DE DECLARACION DE VARIABLES

struct_declaracion_var ::=

struct_declaracion_var COMA IDENTIFICADOR

|struct_declaracion_var COMA IDENTIFICADOR IGUAL struct_operador_logico

|keep_state VAR tipo_dato IDENTIFICADOR IGUAL struct_operador_logico

|keep_state VAR tipo_dato IDENTIFICADOR;

struct_declaracion_var_f ::=

struct_declaracion_var_f COMA IDENTIFICADOR

|struct_declaracion_var_f COMA IDENTIFICADOR IGUAL struct_operador_logico

|VAR tipo_dato IDENTIFICADOR IGUAL struct_operador_logico

|VAR tipo_dato IDENTIFICADOR;

ESTADO KEEP

keep_state ::= KEEP

|;

VALORES DE DATOS

valor ::=

IDENTIFICADOR

|strunt_llamada_funcion

|struct_llamada_arreglo

|especiales_q_retornan

|NUMERO

|DECIMAL

|CADENA

|CHARACTER

|TRUE

|FALSE;

strunt_llamada_funcion ::=

IDENTIFICADOR PA_A state_params PA_C;

parametros ::=

parametros COMA valor_operacion

|valor_operacion;

state_params ::=

parametros

|;

struct_llamada_arreglo ::=

IDENTIFICADOR dim_arreglo;

TIPO DE DATOS

tipo_dato ::=

DOBLE

|ENTERO

|CARACTER_RSV

|CADENA_RSV

|BOOLEAN;

ESTRUCTURA LOGICA

struct_operador_logico ::=

struct_operador_logico OR struct_operador_logico

|struct_operador_logico NOR struct_operador_logico

|struct_operador_logico XOR struct_operador_logico

|struct_operador_logico AND struct_operador_logico

|struct_operador_logico NAND struct_operador_logico

|NOT PA_A struct_operador_logico PA_C

|struct_operador_relacional

|valor_operacion ;

state_op_logicos ::= OR | NOR | XOR | AND | NAND;

struct_operador_relacional ::=

valor_operacion IGUAL IGUAL valor_operacion

| valor_operacion DIFERENTE valor_operacion

| valor_operacion MAYOR_Q valor_operacion

| valor_operacion MENOR_Q valor_operacion

| valor_operacion MAYOR_IGUAL valor_operacion

| valor_operacion MENOR_IGUAL valor_operacion

| IS_NULL valor_operacion;

state_op_racional ::= IGUAL_IGUAL | DIFERENTE | MAYOR_Q | MENOR_Q |

MAYOR_IGUAL | MENOR_IGUAL;

VALOR OPERACIONAL

valor_operacion ::=

valor_operacion MAS valor_operacion

| valor_operacion MENOS valor_operacion

| valor_operacion POR valor_operacion

| valor_operacion DIVISION valor_operacion

| valor_operacion MODULO valor_operacion

| valor_operacion POTENCIA valor_operacion

| PA_A valor_operacion PA_C

| valor;

struct_asig_var ::=

IDENTIFICADOR IGUAL struct_operador_logico

|IDENTIFICADOR MAS_IGUAL struct_operador_logico

|IDENTIFICADOR MAS_MAS

|IDENTIFICADOR MENOS_MENOS

|IDENTIFICADOR error struct_operador_logico;

Estructura arreglo

struct_declaracion_array ::=

struct_declaracion_array COMA IDENTIFICADOR dim_arreglo

|struct_declaracion_array COMA IDENTIFICADOR

|struct_declaracion_array COMA IDENTIFICADOR dim_arreglo IGUAL LLAVE_A

valor_arreglo LLAVE_C

|keep_state VAR tipo_dato ARREGLO IDENTIFICADOR dim_arreglo IGUAL

LLAVE_A valor_arreglo LLAVE_C

|keep_state VAR tipo_dato ARREGLO IDENTIFICADOR

|keep_state VAR tipo_dato ARREGLO IDENTIFICADOR dim_arreglo;

struct_declaracion_array_f ::=

struct_declaracion_array_f COMA IDENTIFICADOR dim_arreglo

|struct_declaracion_array_f COMA IDENTIFICADOR

|struct_declaracion_array_f COMA IDENTIFICADOR dim_arreglo IGUAL

LLAVE_A valor_arreglo LLAVE_C

|VAR tipo_dato ARREGLO IDENTIFICADOR dim_arreglo IGUAL LLAVE_A
valor_arreglo LLAVE_C

|VAR tipo_dato ARREGLO IDENTIFICADOR

|VAR tipo_dato ARREGLO IDENTIFICADOR dim_arreglo;

TAMANIO ARREGLO

dim_arreglo ::=

dim_arreglo CORCHETE_A valor_operacion CORCHETE_C

|CORCHETE_A valor_operacion CORCHETE_C;

VALORES ARREGLO

valor_arreglo ::=

valores_array_mas_dim

|valores_array ;

valores_array_mas_dim ::=

LLAVE_A valores_array_mas_dim LLAVE_C

|valores_array_mas_dim COMA LLAVE_A valores_array_mas_dim LLAVE_C

|LLAVE_A valores_array LLAVE_C

|valores_array_mas_dim COMA LLAVE_A valores_array LLAVE_C;

valores_array ::=

valores_array COMA tipo_valor_op_logico

|tipo_valor_op_logico;

tipo_valor_op_logico ::= struct_operador_logico;

ESTRUCTURA ASIGNACION VALORES ARREGLO

struct_asig_arreglo ::=

IDENTIFICADOR dim_arreglo IGUAL tipo_valor_op_logico;

STRUCUTRA DE UNA FUNCION

struct_funcion ::=

tipo_dato IDENTIFICADOR PA_A struct_parametros PA_C st_ind code_funcion
st_ded
|KEEP tipo_dato IDENTIFICADOR PA_A struct_parametros PA_C st_ind
code_funcion st_ded;

struct_procedimiento ::=

IDENTIFICADOR PA_A struct_parametros PA_C st_ind code_metodo st_ded
|KEEP IDENTIFICADOR PA_A struct_parametros PA_C st_ind code_metodo
st_ded;

struct_parametros ::=

params_lleno

|param_vacio;

params_lleno ::=

struct_parametros COMA tipo_dato IDENTIFICADOR

|tipo_dato IDENTIFICADOR;

param_vacio ::= ;

code_funcion ::=

struct_declaracion_var_f code_funcion

|struct_asig_var code_funcion

|struct_declaracion_array_f code_funcion

|struct_asig_arreglo code_funcion

|struct_llamada_funcion code_funcion

|struct_si code_funcion

|struct_switch code_funcion

|struct_para code_funcion

|struct_mientras code_funcion

|struct_hacer code_funcion

|struct_reproducir code_funcion

|struct_esperar code_funcion

|struct_ordenar code_funcion

|struct_sumarizar code_funcion

|struct_longitud code_funcion

```
|struct_mensaje code_funcion  
  
|RETORNAR struct_operador_logico code_funcion  
  
|RETORNAR error  
  
|;
```

code_metodo ::=

```
    struct_declaracion_var_f code_metodo  
  
    |struct_asig_var code_metodo  
  
    |struct_declaracion_array_f code_metodo  
  
    |struct_asig_arreglo code_metodo  
  
    |struct_llamada_funcion code_metodo  
  
    |struct_si code_metodo  
  
    |struct_switch code_metodo  
  
    |struct_para code_metodo  
  
    |struct_mientras code_metodo  
  
    |struct_hacer code_metodo  
  
    |struct_reproducir code_metodo  
  
    |struct_esperar code_metodo  
  
    |struct_ordenar code_metodo  
  
    |struct_sumarizar code_metodo  
  
    |struct_longitud code_metodo  
  
    |struct_mensaje code_metodo  
  
    |;
```

Funciones reservadas

struct_reproducir ::=

REPRODUCIR PA_A notas COMA valor_operacion COMA valor_operacion
COMA valor_operacion PA_C;

struct_esperar ::=

ESPERAR PA_A valor_operacion COMA valor_operacion PA_C;

struct_ordenar ::=

ORDENAR PA_A IDENTIFICADOR COMA tipo_orden PA_C;

struct_sumarizar ::=

SUMARIZAR PA_A IDENTIFICADOR PA_C;

struct_longitud ::=

LONGITUD PA_A valor_operacion PA_C;

struct_mensaje ::=

MENSAJE PA_A mensajes PA_C

|MENSAJE PA_A PA_C;

mensajes ::=

mensajes MAS struct_operador_logico

|struct_operador_logico;

NOTAS MUSICALES

```
notas ::= DO | RE | MI | FA | SOL | LA | SI  
        |DO_S | RE_S | FA_S | SOL_S | LA_S;
```

OCTAVAS

```
//octavas ::=
```

TIPO DE ORDEN

```
tipo_orden ::= ASCENDENTE | DESCENDENTE | PARES | IMPARES | PRIMOS ;
```

ORDENACION

```
orden_valor ::= valor_operacion  
               |valor_arreglo;
```

STRUTURA PRINCIPAL MEOTODO

```
struct_principal ::= PRINCIPAL PA_A PA_C st_ind code_metodo st_ded;
```

ESTRUTURA SI-SINO SI-SINO

```
struct_si ::=
```

SI PA_A struct_operador_logico PA_C st_ind code_funcion st_ded struct_sino;

struct_sino ::=

SINO SI PA_A struct_operador_logico PA_C st_ind code_funcion st_ded

struct_sino

|SINO st_ind code_funcion st_ded

;

ESTRUCTURA SWITCH

struct_switch ::=

SWITCH PA_A IDENTIFICADOR PA_C st_ind struct_caso st_ded;

struct_caso ::=

CASO valor_operacion st_ind code_funcion caso_salir st_ded struct_caso

|DEFAULT st_ind code_funcion st_ded

;

caso_salir ::= SALIR

;

ESTRUCTURA PARA

struct_para ::=

PARA PA_A para_asig_var PUNTO_COMA condicion_para PUNTO_COMA
incremento_para PA_C st_ind code_funcion st_ded;

para_asig_var ::=

VAR tipo_dato IDENTIFICADOR IGUAL valor_operacion

|IDENTIFICADOR IGUAL valor_operacion;

condicion_para ::= struct_operador_logico;

incremento_para ::=

IDENTIFICADOR IGUAL valor_operacion

|IDENTIFICADOR MAS_IGUAL valor_operacion

|IDENTIFICADOR MAS_MAS

|IDENTIFICADOR MENOS_MENOS;

Estructura mientras, hacer-mientras,

struct_mientras ::=

MIENTRAS PA_A struct_operador_logico PA_C st_ind code_funcion st_ded;

struct_hacer ::=

HACER st_ind code_funcion state_continuar st_ded MIENTRAS PA_A

struct_operador_logico PA_C;

state_continuar ::= CONTINUAR

|;

especiales_q_retornan ::=

 struct_reproducir

 |struct_sumarizar

 |struct_longitud;