Parser - C-

Diseño de compiladores - Doc. Víctor Manuel de la Cueva

José Manuel Beauregard Méndez
A01021716

La librería ply es requerida para compilar este proyecto

Expresiones regulares

Expresión Regular	Token
\d+	NUM
[a-zA-Z_][a-zA-Z0-9_]*	ID
\/*(*(?!\/)|[^*])**\/	COMMENT
“=“	EQUAL
\+	PLUS
-	MINUS
*	TIMES
/	DIVIDE
<	LESS
>	GREATER
;	SEMICOLOM
,	COMMA
\(	LPAREN
\)	RPAREN
\[	LBRACKET
\]	RBRACKET
\{	LBLOCK
}	RBLOCK
\$	ENDFILE

Gramática

Rule 0 S' -> start

Rule 1 start -> declaration_list

Rule 2 declaration_list -> declaration_list declaration

Rule 3 declaration_list -> declaration

Rule 4 declaration -> var_declaration

Rule 5 declaration -> fun_declaration

Rule 6 declaration -> ENDFILE

Rule 7 var_declaration -> type_specifier ID SEMICOLON

Rule 8 var_declaration -> type_specifier ID LBRACKET NUMBER RBRACKET SEMICOLON

Rule 9 type_specifier -> INT

Rule 10 type_specifier -> VOID

Rule 11 fun_declaration -> type_specifier ID LPAREN params RPAREN compound_stmt

Rule 12 params -> param_list

Rule 13 params -> VOID

Rule 14 param_list -> param_list COMMA param

Rule 15 param_list -> param

Rule 16 param -> type_specifier ID

Rule 17 param -> type_specifier LBRACKET RBRACKET

Rule 18 compound_stmt -> LBLOCK local_declarations statement_list RBLOCK

Rule 19 local_declarations -> local_declarations var_declaration

Rule 20 local_declarations -> <empty>

Rule 21 statement_list -> statement_list statement

Rule 22 statement_list -> <empty>

Rule 23 statement -> expression_stmt

Rule 24 statement -> compound_stmt

Rule 25 statement -> selection_stmt

Rule 26 statement -> iteration_stmt

Rule 27 statement -> return_stmt

Rule 28 expression_stmt -> expression SEMICOLON

Rule 29 expression_stmt -> SEMICOLON

Rule 30 selection_stmt -> IF LPAREN expression RPAREN statement

Rule 31 selection_stmt -> IF LPAREN expression RPAREN statement ELSE statement

Rule 32 iteration_stmt -> WHILE LPAREN expression RPAREN statement

Rule 33 return_stmt -> RETURN SEMICOLON

Rule 34 return_stmt -> RETURN expression SEMICOLON

Rule 35 expression -> var EQUAL expression

Rule 36 expression -> simple_expression

Rule 37 var -> ID

Rule 38 var -> ID LBRACKET expression RBRACKET

Rule 39 simple_expression -> additive_expression relop additive_expression

Rule 40 simple_expression -> additive_expression

Rule 41 relop -> LE

Rule 42 relop -> LT

Rule 43 relop -> GREATER

Rule 44 relop -> LESS

Rule 45 relop -> COMPARE

Rule 46 relop -> NE

Rule 47 additive_expression -> additive_expression addop term

Rule 48 additive_expression -> term

Rule 49 addop -> PLUS

Rule 50 addop -> MINUS

Rule 51 term -> term mulop factor

Rule 52 term -> factor

Rule 53 mulop -> TIMES

Rule 54 mulop -> DIVIDE

Rule 55 factor -> LPAREN expression RPAREN

Rule 56 factor -> ID

Rule 57 factor -> call

Rule 58 factor -> NUMBER

Rule 59	call -> ID LPAREN args RPAREN
Rule 60	args -> arg_list
Rule 61	args -> <empty>
Rule 62	arg_list -> arg_list COMMA expression
Rule 63	arg_list -> expression