Cairo University

Faculty of Engineering

Computer Department – Forth Year

Compilers Project

Submitted to:

Eng. Yousef Ghattas

Submitted by:

Omar Adel

(SEC : 2 - BN : 3)

Moutaz Samir

(SEC : 2 - BN : 19)

Ramadan Ahmed

(SEC : 1 - BN : 23)

Tarek Mohamed

(SEC : 1 - BN : 28)

Project Overview

A programming language that supports basic data types (int) and supports also assignment statements and mathematical and logical expressions, it also provides control statements and conditional statements.

Tools and Technologies Used

- Python
- Windows (Operating System)
- C# WinForms (Graphical User Interface GUI)

List of Tokens

Token	Description
INT	Integer data type
CONST	Constant data type
IF	If keyword (conditional)
ELSE	Else keyword
FOR	For loop
WHILE	While loop
DO	Do keyword
SWITCH	Switch keyword
CASE	Case (option in a switch statement)
BREAK	Break keyword
DEFAULT	Default (default case in switch statement)
TRUE	True keyword (not equal to zero)
FALSE	False keyword (equal to zero)
AND	Logical and
OR	Logical or
NOT	!
EQUAL	==
NOT_EQUAL	!=
LESS_ EQUAL	<=
GREATER_ EQUAL	>=
LESS	<
GREATER	>
COLON	:
COMMA	,
ASSIGN	=
SEMI_COLON	;
L_PAREN	(
R_PAREN	)
L_BRACE	{
R_BRACE	}
PLUS	+
MINUS	-
MULTIPLY	*
DIVIDE	/

Production Rules

- **type_spec**
  - INTEGER_TYPE
- **Program**
  - statement_list

- **statement_list**
  - statement
  - statement statement_list
- **statement**
  - BREAK SEMI_COLON
  - | selection_statement
  - | iteration_statement
  - | assignment_statement SEMI_COLON
  - | declaration_statement SEMI_COLON
  - | const_statement SEMI_COLON
  - | SEMI_COLON
- **init_for**
  - assignment_statement
  - | declaration_statement
- **compound_statement**
  - L_BRACE statement_list R_BRACE
  - | L_BRACE R_BRACE
- **assignment_statement**
  - variable ASSIGN expr
- **selection_statement**
  - IF L_PAREN boolean_expression R_PAREN compound_statement
  - | IF L_PAREN boolean_expression R_PAREN compound_statement ELSE compound_statement
  - | SWITCH L_PAREN variable R_PAREN L_BRACE (CASE (INTEGER_VALUE | MINUS INTEGER_VALUE) COLON compound_statement)+ (DEFAULT COLON compound_statement) R_BRACE
- **iteration_statement**
  - WHILE L_PAREN boolean_expression R_PAREN compound_statement
  - | DO compound_statement WHILE L_PAREN boolean_expression R_PAREN SEMI_COLON
  - | FOR L_PAREN init_for SEMI_COLON boolean_expression SEMI_COLON assignment_statement R_PAREN compound_statement
- **variable_declaration**
  - TYPE variable ASSIGN expr
  - TYPE variable
- **const_declaration**
  - CONSTANT TYPE variable ASSIGN expr
- **expr**
  - term ((PLUS | MINUS) term)*

- **term**
  - factor ((MULTIPLY | DIVIDE) factor)*
- **factor**
  - PLUS factor
  - | MINUS factor
  - | INTEGER_VALUE
  - | LPAREN expr RPAREN
  - | variable
- **boolean_expression**
  - boolean_term (relation boolean_term)*
- **boolean_term**
  - NOT boolean_term
  - | TRUE
  - | FALSE
  - | L_PAREN boolean_expression R_PAREN

List of Quadruples

Quadruple	Description
(=, a , ,b)	b=a
(equal, a,b,Rx)	If (a ==b) Rx = true and false otherwise
(jfalse, Lx, Rx,)	Jump to Lx if Rx is false
(jtrue, Lx, Rx,)	Jump to Lx if Rx is true
(uminus, a, ,Rx)	Rx = -a
(+, a, b, Rx)	Rx = a + b
(-, a, b, Rx)	Rx = a - b
(*, a, b, Rx)	Rx = a * b
(/, a, b, Rx)	Rx = a / b
(AND, a, b, Rx)	Rx = a AND b

(OR, a, b, Rx)	Rx = a OR b
(Greater, a, b, Rx)	Rx = true if a > b and false otherwise
(LESS, a, b, Rx)	Rx = true if a < b and false otherwise
(jmp, Lx, ,)	Jump to Lx