LAB REPORT

Course No: CSE 3212

Course Title: Compiler Design Laboratory

Submitted to:

Dola Das
Assistant Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology (KUET)

Dipannita Biswas
Lecturer
Department of Computer Science and Engineering
Khulna University of Engineering & Technology (KUET)

Submitted by:

Simon Walter Tudu
1807121
3rd Year
2nd Semester
Department of Computer Science and Engineering

Submission Date: 20 December 2022

Introduction

Flex : Flex is a tool for generating scanners. Flex source is a table of programfragments . Flex generates lex.yy.c which defines a routine yylex( ).

Flex input file format is given below:
{ definitions }
%%
{ rules }
%%
{user subroutines}

Bison : Bison is a general purpose parser generator that converts a grammardescription for an LALR context-free grammar into a c program to parse that grammar.

Bison input file format is given below :
%{
C declarations { types , variables , functions , preprocessor commands}
%}
/* Bison Declarations (grammar symbols , operator precedence declaration ,attribute data type) */
%%
/* grammar rules go here */
%%
/* additional c code goes here */

Necessary Equipment

Command Prompt , Laptop , Sublime Text Editor ,Flex , Bison.

Run the program in terminal

- bison -d 1807121.y
- flex 1807121.l
- gcc 1807121.tab.c lex.yy.c -o out
- o

Procedure

- The code is divided into two parts: flex file (.l) and bison file (.y) .
- Input expression checks the lex (.y) file and if the expression satisfies therule then it checks the CFG into the bison file .
- It's a bottom up parser and the parser constructs the parse tree firs ,matchesthe leaves node with the rules and if the CFG matches then it gradually goes tothe root .
- There is an input file called "a.txt" from which inputs are gained.

Description of my Project

It is known that A lexical analyzer divides the input into meaningful units . Fora C program the units are variables , constants , keyboards , operators , punctuation etc . These units are also called tokens. Parser finds the relationship between input tokens . For a C program , one needs to identify valid expressions, statements , blocks , procedures etc .

The Tokes that I have used in my compiler project are given below :

Token	Description
COMMA	Returned when “,” pattern is found
SEMICOLON	Returned when “;” pattern is found
COLON	Returned when “:” pattern is found
SHOW_VAR	Returned when “show_var” pattern is found
SHOW_STR	Returned when “show_str” pattern is found
SHOW_LINE	Returned when “show_line” pattern is found
SCAN	Returned when “scan” pattern is found
NUM	Returned when “[0-9]” pattern is found
MAIN	Returned when “main_T” pattern is found
INT	Returned when “int_T” pattern is found
DOUBLE	Returned when “double_T” pattern is found
FLOAT	Returned when “float_T” pattern is found
CHAR	Returned when “char_T” pattern is found
IF	Returned when “if_T” pattern is found
ELSE	Returned when “else_T” pattern is found
ELSEIF	Returned when “eif_T” pattern is found
SWITCH	Returned when “switch_T” pattern is found
DEFAULT	Returned when “default_T” pattern is found
FBS	Returned when “fbs_T” pattern is found
FBE	Returned when “fbe_T” pattern is found
SBS	Returned when “sbs_T” pattern is found
SBE	Returned when “sbe_T” pattern is found
FOR	Returned when “for_T” pattern is found
TO	Returned when “To” pattern is found
INCREMENT_BY	Returned when “inc_by” pattern is found
WHILE	Returned when “while_T” pattern is found

GT	Returned when “greater_T” pattern is found
LT	Returned when “less_T” pattern is found
GE	Returned when “greater_eT” pattern is found
LE	Returned when “less_eT” pattern is found
NE	Returned when “not_equal_T” pattern is found
EQ	Returned when “equal_T” pattern is found
LESS	Returned when “LESS” pattern is found. Used with while loop for decrementing order
GREAT	Returned when GREAT” pattern is found. Used with while loop for incrementing order
ASSGN	Returned when “assign_T” pattern is found
PLUS	Returned when “add_T” pattern is found
MINUS	Returned when “sub_T” pattern is found
MUL	Returned when “mul_T” pattern is found
DIV	Returned when “div_T” pattern is found
MODULUS	Returned when “mod_T” pattern is found
POWER	Returned when “power_T” pattern is found

Precedence of the operators: The list of the operators according to presidencyfrom higher to lower are given below:

- < > :no associativity
- IF ELSE :no associativity
- All logical operators :left associativity
- * / :left associativity
- + - :left associativity
- =/==/!= :right associativity

Reference

- Flex (Fast Lexical Analyzer Generator) - https://www.geeksforgeeks.org/flex-fast-lexical-analyzer-generator/
- Introduction to YACC - https://www.geeksforgeeks.org/introduction-to-yacc/
- Flex - https://en.wikipedia.org/wiki/Flex_(lexical_analyser_generato