

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UCS1602 - Compiler Design

EX - 6 : Implementation of Syntax checker using Lex and Yacc Tools

NAME : Gayathri M
REG NO : 185001050
DATE : 29/03/2021

Program Code:

syntax.l

```
%{
#include<stdio.h>
#include<stdlib.h>
#include "y.tab.h"
void yyerror(char *);
extern int yylval;
%}

stringid ([_a-zA-Z][_a-zA-Z0-9]*\[[0-9]*\])
num ([0-9\.]+)
string (\".*\")
character (\'.*\')

%%

("++"|"--") {return UNARY;}
("<"| "<="| ">"| ">="| "=="| "!=") { return RELOP; }
("+"| "-"| "*"| "/"| "%"| "<<"| ">>"| "!"| "&&"| "||") { return OP; }
("+="| "-="| "*="| "/="| "=") { return ASSIG_OP; }
```

```

("int"|"float"|"char"|"double") { return DTYPE; }
if {return IF;}
else {return ELSE;}
while {return WHILE;}
for {return FOR;}

[_a-zA-Z][_a-zA-Z0-9]* { return ID; }
{stringid} { return STRING_ID; }
{num} { return NUM; }
{string} { return STRING;}
{character} { return CHAR;}
[ \t\n]+ { }
(";"|"\".|"("|")"|"{"|"}") {return *yytext;}
. {
    fprintf(stderr,"Unknown token found: <%s>\n", yytext);
}
%%

```

syntax.y

```

%{
    #include<stdio.h>
    #include<stdlib.h>
    #include<math.h>
    #include"y.tab.h"
    int yylex(void);
    void yyerror();
    int flag=0;
}%

%token NUM STRING CHAR ID STRING_ID OP ASSIG_OP DTYPE RELOP IF ELSE
      WHILE FOR UNARY

%%
program: statement program
        | statement

```

```

statement: declaration';'
          | assignment';'
          | expr';'
          | conditional
          | loop

declaration: DTYPE ID
            | DTYPE STRING_ID
            | DTYPE ID ASSIG_OP expr
            | DTYPE STRING_ID ASSIG_OP expr
assignment: ID ASSIG_OP expr
conditional: IF expr statement ELSE statement
            | IF expr statement ELSE '{' program '}'
            | IF expr '{' program '}' ELSE statement
            | IF expr '{' program '}' ELSE '{' program '}'
            | IF expr statement
            | IF expr '{' program '}'

loop: WHILE '(' expr ')' '{' program '}'
     | WHILE '(' assignment ')' '{' program '}'
     | FOR '(' DTYPE ID ASSIG_OP expr ';' expr ';' expr ')'
     | FOR '(' DTYPE ID ASSIG_OP expr ';' expr ';' expr ')' '{'
program '}'
     | FOR '(' ID ASSIG_OP expr ';' expr ';' expr ')' statement
     | FOR '(' ID ASSIG_OP expr ';' expr ';' expr ')' '{' program '}'

expr: expr OP expr
     | expr RELOP expr
     | '(' expr ')'
     | ID UNARY
     | ID | NUM | STRING | CHAR

%%

void yyerror(){
    fprintf(stderr, "Invalid syntax\n");
    flag = 1;
    return;
}

int yywrap(){
    return 1;}

```

```
int main()
{
    yyparse();
    if(!flag){
        printf("Valid syntax\n");
    }
    return 0;
}
```

input.txt

```
char s[50] = "xyz";
char c = 'a';
b=2;
a=4*3;
char s[50];
char c;
s = "abc";
c = 'x';
if(a>b)
    a+=2;
else if(b>c)
{
    c++;
}
else
    c+=2;
while(i<10)
{
    i++;
}
for(int x=1; x<10; x++)
{
    a--;
}
```

Output Snapshots:

```
msml@MSMLs-MacBook-Pro ex6 % lex syntax.l
msml@MSMLs-MacBook-Pro ex6 % yacc -d syntax.y
conflicts: 6 shift/reduce
msml@MSMLs-MacBook-Pro ex6 % gcc y.tab.c lex.yy.c
msml@MSMLs-MacBook-Pro ex6 % ./a.out < input.txt
Valid syntax
msml@MSMLs-MacBook-Pro ex6 % █
```

Learning Outcomes :

- I learnt to install and use yacc tool.
- I learnt the necessity of a syntax checker.
- I learnt how a syntax checker works.
- I learnt to implement syntax checker using yacc and lex tools.
- I learnt how to create and add rules to check for given grammars.