Department of Computer Science and Engineering

S.G.Shivanirudh , 185001146, Semester VI

29 March 2021

UCS1602 - Compiler Design

Exercise 6:Implementation of Syntax Checker using YaccTool

Objective:

Develop a Syntax checker to recognize the tokens necessary for the following statements by writing suitable grammars Assignment statement Conditional statement Looping statement

Code:

Lex:

```
%{
    #include<stdio.h>
    #include "y.tab.c"
    extern YYSTYPE yylval;
%}

kw int|char|float|double|while|do
if if
else else
for for
num [0-9]+
id [a-z][a-z]*

%%
{num} {return NUM;}
{kw} {return KW;}
{if} {return IF;}
{for} {return FOR;}
{else} {return ELSE;}
"(" {return POPEN;}
")" {return PCLOSE;}
"{" {return BOPEN;}
"}" {return BCLOSE;}
{id} {return ID;}
("+="|"-="|"*="|"/="|"=") {return AOP;}
("++"|"--") {return CHANGE_OP;}
("=="|"!="|">"|"<"|">="|"<=") {return ROP;}
";" {return SEP;}
[+\-^*/,().] {return *yytext;}
[\t]
[ ]
[\n]

. return yytext[0];
%%
int yywrap(){
    return 1;
}
```

Yacc:

```
%{
```

```
2     #include<stdio.h>
3     #define YYSTYPE double
4
5     int flag = 0;
6     int yylex(void);
7 %}
8
9 %token NUM ID KW AOP
10 %token IF ELSE ROP
11 %token POPEN PCLOSE BOPEN BCLOSE
12 %token FOR WHILE
13 %token SEP
14 %token CHANGE_OP
15
16
17 %%
18 stmt : assn_stmt
19   | cond_stmt
20   | loop_stmt
21 ;
22 assn_stmt : ID AOP expr {printf("\nAssignment statement found
       \n");}
23 ;
24 expr : expr '+' expr
25   | expr '-' expr
26   | expr '*' expr
27   | expr '/' expr
28   | NUM
29   | ID
30 ;
31
32 cond_stmt : IF cond stmt continue {printf("\nConditional
       statement found\n");}
33 ;
34 cond : POPEN rel_expr PCLOSE
35 ;
36 continue : ELSE stmt
37          |
38 ;
39 rel_expr : expr ROP expr
40 ;
41
42 loop_stmt : for_stmt
43           | while_stmt
44 ;
```

```
45 for_stmt : FOR POPEN assn_stmt SEP rel_expr SEP inc_expr
       PCLOSE BOPEN stmt BCLOSE {printf("\nLooping statement
       found\n");}
46 ;
47
48 inc_expr :  assn_stmt
49             | expr CHANGE_OP
50 ;
51 while_stmt : WHILE cond BOPEN stmt BCLOSE
52 ;
53
54
55 %%
56
57 int yyerror (char const* s)
58 {
59     printf("\nSyntactically Incorrect: %s\n", s);
60     flag=1;
61 }
62
63 int main(int argc, char **argv){
64     if(argc != 2){
65         fprintf(stderr, "Enter file name as argument!\n");
66         return 1;
67     }
68     yyin = fopen(argv[1], "rt");
69     if (!yyin){
70         fprintf(stderr, "File not found!\n");
71         return 2;
72     }
73     yyparse();
74     if(flag==0)
75         printf("\nSyntactically correct\n");
76     return 0;
77 }
```

Output:

Correct syntax:

```
for(i = 0;i < 10; i++){
    if(x < 10)
        x += 8
    else
        y -= 9
}
```

```
Assignment statement found
Assignment statement found
Assignment statement found
Conditional statement found
Looping statement found
Syntactically correct
```

Incorrect syntax:

```
for(i = 0;i < 10; i++){
    if(x < 10)
        x += 8
    else
        y -= 9;
}
```

```
Assignment statement found

Assignment statement found

Assignment statement found

Conditional statement found

Syntactically Incorrect: syntax error
```

Learning Outcomes:

- Understood the basic concept of Syntax Checker.
- Learnt how to identify control structures using yacc and lex.
- Learnt to use yacc efficiently for specifying grammar.
