

Compiler Construction Experiment 06

Shubham Golwal | 2020300015 | TE COMPS

12-04-23

AIM: To perform lexical analysis.

THEORY:

Syntax Directed Translation has augmented rules to the grammar that facilitate semantic analysis. SDT involves passing information bottom-up and/or top-down to the parse tree in form of attributes attached to the nodes. Syntax-directed translation rules use

- 1) lexical values of nodes,
- 2) constants &
- 3) attributes associated with the non-terminals in their definitions.

The general approach to Syntax-Directed Translation is to construct a parse tree or syntax tree and compute the values of attributes at the nodes of the tree by visiting them in some order. In many cases, translation can be done during parsing without building an explicit tree.

Syntax-directed translation is a process of translating a program from one language to another using a syntax-directed definition of the source language. Here are some of the advantages of syntaxdirected translation:

Helps in generating efficient code: Syntax-directed translation allows for the automatic generation of code from the source language to the target language, which can help in generating efficient code. By using a syntax-directed definition, it is possible to optimize the code generated for specific hardware platforms and make it more efficient.

Increases productivity: Syntax-directed translation reduces the amount of time and effort required to write code by automating the process of translating code from one language to another. This can help developers focus more on designing and developing software rather than worrying about the details of the syntax.

Improves code quality: Syntax-directed translation can help in detecting errors and inconsistencies in the code. By using a syntaxdirected definition, it is possible to ensure that the generated code is correct and conforms to the specification of the target language.

Provides better language integration: Syntax-directed translation allows for seamless integration between different programming languages. By translating code from one language to another, it is possible to use libraries and frameworks written in different languages, which can help in developing complex applications.

Overall, syntax-directed translation is a useful technique for automating the process of translating code from one language to another, improving code quality, and increasing productivity.

CODE:

Exp.l

```
%{
#include <stdio.h>
#include "y.tab.h"
#include <string.h>
int variable_one = 0;
int variable_two = 0;
}%

%%
[nsew] {
yylval.intval = *yytext;
return DIRECTION;
}
[ \t] ;
"START" {return START;}
"$" {return 0;}
\n|. return yytext[0];

%%
```

calc.y

```
%{
#include <string.h>
extern int yylex();

void yyerror(char *s);
    int posX;
    int posY;
    int flag = 0;
}%

%union {
    int intval;
}

%token<intval> DIRECTION
%token<intval> START

%%
```

```
statement: statement direction  
| START {posX = 0; posY = 0;}
```

```
direction: DIRECTION {
```

```
    switch($1){  
        case 110:  
            posY += 1;  
            break;  
        case 119:  
            posX -= 1;  
            break;  
        case 115:  
            posY -= 1;  
            break;  
        case 101:  
            posX += 1;  
            break;  
    }  
}
```

```
%%
```

```
void yyerror(char *s){  
    printf("The given string is Invalid");  
    flag = 1;  
    return;  
}
```

```
int main(){  
    yyparse();  
    if(!flag) printf("Final position is = (%d,%d)", posX, posY);  
    else printf("Please try again");  
}
```

Output:

```
students@CE-Lab4-606-U09: ~/Downloads/Downl
students@CE-Lab4-606-U09:~/Downloads/Downl$ yacc -d yacc.y
students@CE-Lab4-606-U09:~/Downloads/Downl$ cc lex.yy.c y.tab.c
students@CE-Lab4-606-U09:~/Downloads/Downl$ ./a.out
knews
X = 0,Y = 1
X = 1,Y = 1
X = 0,Y = 1
X = 0,Y = 0
students@CE-Lab4-606-U09:~/Downloads/Downl$ ./a.out
knewsbbc
X = 0,Y = 1
X = 1,Y = 1
X = 0,Y = 1
X = 0,Y = 0
Terminal not found.
students@CE-Lab4-606-U09:~/Downloads/Downl$ ./a.out
kneennw
X = 0,Y = 1
X = 1,Y = 1
X = 2,Y = 1
X = 2,Y = 2
X = 2,Y = 3
X = 1,Y = 3
students@CE-Lab4-606-U09:~/Downloads/Downl$
```

CONCLUSION:

- In this experiment, We understood the concept of syntax directed translation.
- Understood how to construct the SDT parsing tree and implemented the code for the same using the yacc tool.
- Thus, we successful implemented program to perform Syntax directed translation using yacc tool.