How to Code in YACC

CS321 : Compiler Lab

13-Feb-2025

YACC is a tool used to generate parsers for context-free grammars. It is typically used with Lex (a lexical analyzer) to process structured input. YACC generates a C program that can parse input based on a specified grammar.

1 Installation

To use YACC, you must first install BISON. Use the following command to install BISON on a Linux system.

```
sudo apt-get install bison
```

2 YACC Workflow

- Use Lex (Flex) to generate tokens.
- Define tokens in YACC and create grammar rules.
- Compile Lex and YACC output.
- Run the parser on an input file.

3 YACC File Structure

A YACC source file (`.y`) consists of three sections:

1. **Declarations Section**: Token and type declarations.
2. **Rules Section**: Grammar rules and associated C actions.
3. **User Code Section**: Supporting C code (such as the `main` function).

4 Example

4.1 Arithmetic Expression Parser (`calc.y`)

```
%{
/* Definition section */
#include<stdio.h>
int flag=0;
```

```
%}

%token NUMBER
%left '+' '-'
%left '*' '/' '%'
%left '(' ')'

/* Rule Section */
%%

ArithmeticExpression: E{
                printf("Result=%d\n", $$);
                return 0;
                };

E:E'+'E {$$=$1+$3;}
 |E'-'E {$$=$1-$3;}
 |E'*'E {$$=$1*$3;}
 |E'/'E {$$=$1/$3;}
 |E'%'E {$$=$1%$3;}
 |'('E')' {$$=$2;}
 | NUMBER {$$=$1;}
 ;

%%

/* Use Code Section */
void main() {
    printf("Enter Arithmetic Expression: ");
    yyparse();
    if(flag==0)
        printf("Entered arithmetic expression is Valid.\n");
}

void yyerror()  {
    printf("Entered arithmetic expression is Invalid.\n");
    flag=1;
}
```

4.2 Writing the Lex File (`calc.l`)

```
%{
/* Definition section */
#include<stdio.h>
#include "y.tab.h"
extern int yylval;
%}

/* Rule Section */
```

```
%%
[0-9]+ {
         yylval=atoi(yytext);
         return NUMBER;
        }
[\t] ;
[\n] return 0;
. return yytext[0];
%%

int yywrap() {
    return 1;
}
```

4.3 Steps to Run YACC Program

1. Generate the C code from lex file.
   `lex calc.l`.
2. Generate the C code from yacc file.
   `yacc -d calc.y`.
3. Compile the C code using a C compiler.
   `gcc lex.yy.c y.tab.c -o calc -ll`
4. Execute the compiled program.
   `./calc`

4.4 Sample Input/Output

Input:
Enter Arithmetic Expression: 3+5*2
Output:
Result=13