DATE:

SPCC Lab-07

Aim:- Programming in yacc tool

Theory:- YACC is a computer program for linux OS. It is a LALR parser generator generating a power the most part of compiler that tries to make syntactic sense of source code

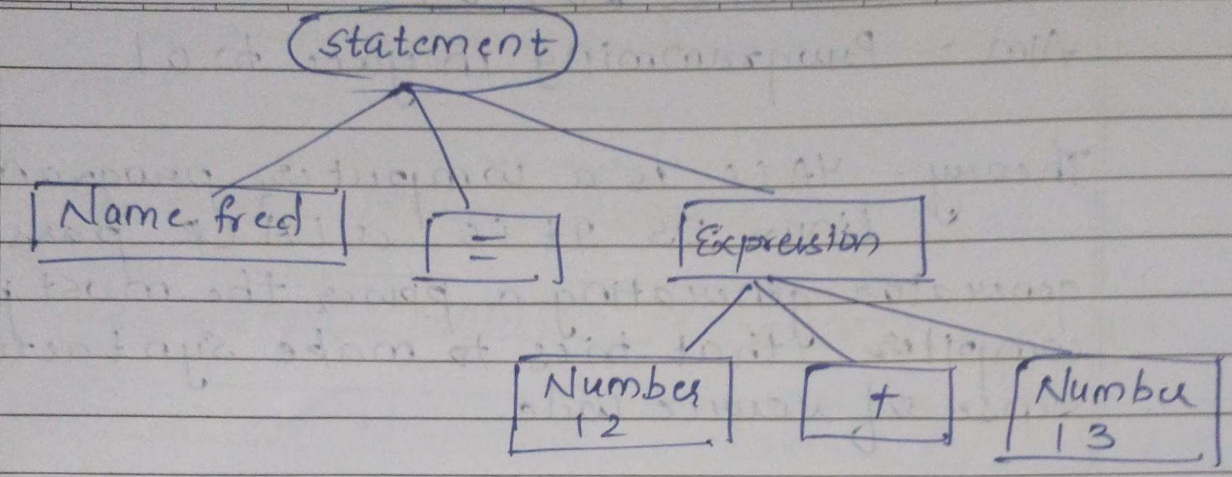
Lex divides the input stream into pieces (tokens) and then yacc takes those pieces and groups them together logically. Its output is a shift reduce parser in C that executes the C snippess associated with each rule as soon as the rule is recognized

Statement → NAME = expression
Expression → NUMBER + NUMBER | NUMBER - NUMBER

The Vertical Bar '|' means there are 2 possibilities for the same symbol i.e an expression can be either an addition or a subtraction. The symbol to the left of → is known as left hand side of rule and symbol to the righ is right hand side symbols that actually appear in the input and are returned by lexer are terminals symbols tokens, while those that appear on the left hand side of some rule are non-terminals

FOR EDUCATIONAL USE

DATE:



'12 + 13' is an expression and fred = expression is a statement

you cannot parse it

- there's ambiguous grammar
- grammars that need more than one token of look ahead to tell whether it has matched a Rule.

YACC grammar has the same three part structure as a lex specification

The definition section - include declaration of tokens used in grammar, the type of values used on parser table and other odd & ends

The Rules section - This simply consist of a list of grammar rules in pretty much the same format as we used above since ASCII doesn't have → key we use colon between left and right hand side of a rule & put semicolon at the end of each

DATE:

rule

%token NAME NUMBER
%%
statement : NAME = expression | expression ;

expression: NUMBER + NUMBER | NUMBER - NUMBER

unlike lex yacc, pays no attention to line boundaries in rule section

Steps for storing and executing yacc file

1. Write lex file and save it with dot extension
2. write yacc file and save it with dot y extension and filename must be same as that used for lex file
3. Compile the lex file
4. Use bison -dy filename.y to compile yacc file
5. To generate executable file write following command - gcc lex.yy.c y.tab.c filename.exe
6. To run the program - filename.exe