DESIGN LAB (CS69201)

Part I : Lex and Yacc Tutorial

Objective:

The main objective of this tutorial are as follows :

1. To understand the basics of regular expressions and context free grammars.
2. To understand the basics of lexical analysis (tokenization) and use of Lex.
3. To understand the basics of syntax analysis (parsing) and use of Yacc.

Regular Expressions : A regular expression(regex) is a sequence of characters that define a search pattern, mainly for use in pattern matching with strings, or string matching.

Example : Regular expression for searching any alphabet in a sentence irrespective of cases(lowercase or uppercase) is

[a-zA-Z]+

Here, [set-of-characters] matches any character present in the box braces. 'a-z' defines all characters between a to z. Same for 'A-Z'. '+' denotes the set of characters repeated at least one time.

For the sentence, 'Happy New Year', the regular expression defined above searches and detects 'Happy', 'New' and 'Year' from it.

Python has its library for regular expressions known as Python regex (re module). For more, go to this link for Reference. For a video tutorial of regex, go to this link.

Context Free Grammars : Context Free Grammar is formal grammar, the syntax or structure of a formal language can be described using context-free grammar (CFG), a type of formal grammar. The grammar has four tuples: (V,T,P,S).

1. V - It is the collection of variables or nonterminal symbols.
2. T - It is a set of terminals.

3. P - It is the production rules that consist of both terminals and nonterminals.
4. S - It is the Starting symbol.

Example : A-> aAa|bAb|a|b|e gives all palindrome strings of alphabets {a,b}. e is an empty string.

Here , V = {A}, T={a,b}, P = A->aAa|bAb|a|b|e and A is the starting symbol (S).

For more, see this. For more examples, see this.

Lex and Yacc :

In this tutorial, we discuss lex and yacc. We use the Python Lex Yacc(PLY) package for the same here and also in our assignments. The links of the PLY packages are associated below :

1. Dabeaz Github PLY
2. Dabeaz PLY 3.11
3. PLY Download

Lex: Lex is a tool or a computer program that generates Lexical Analyzers (converts the stream of characters into tokens). The Lex tool itself is a compiler. The Lex compiler takes the input and transforms that input into input patterns. Read this for a detailed explanation.

Yacc : Yacc is a lookahead left-to-right rightmost derivation (LALR) parser generator, generating a LALR parser (the part of a compiler that tries to make syntactic sense of the source code) based on a formal grammar. This is run after Lex. Read this for Parsing basics and detailed explanation of Yacc.

Here is a video tutorial illustrating the use of Calculator with the help of Lex and Yacc. The link is : PLY Calculator Video Tutorial.

After completing the tutorial, one should be able to :

1. Get acquainted with the objectives mentioned above.
2. Able to perform standard Context Free Grammar based Parsing.
3. Able to perform web crawling activities using PLY library.

Here is a sample question you can try after going through the tutorial :

Question:

Implement the calculator as below, and terminate when the user inputs the ‘exit’ command:

user: 3 * (4.5 + 5.5)

calc: 30

user: x = 3 * (4.2 + 5.8)

user: y = 5

user: x

calc: 30

user: y

calc: 5

user: x + 2*y

calc: 40

user: x+2*y == 3*x-10*y

calc: yes

user: x+2*y+z

calc: invalid input

user: postfix(a+b*c)

calc: abc*+

user: prefix(a+b*c)

calc: +a*bc

user:exit

calc:exiting...

Instructions:

1. Operator supported: +,-,*,/,%,^,unary minus,==,=.

2. Two special keyword postfix & prefix to get postfix & prefix form of a infix expression.

3. Parenthesis supported.

4. Input can also be float.

5. Give a proper error message in case of invalid/wrong input.