

Page: 6
Date: 11

Assignment - B5

Title : YACC program to implement syntactic analysis

Date of Completion: 11/11/2018

Problem Statement:

Write a program using YACC specifications to implement syntax analysis phase of compiler to recognise simple & compound sentences given in input file.

Objectives :

To understand theories behind parser

To understand automation of parser

To be able to use YACC generators

To understand how instructions are matched & syntax is understood by machine

Outcome :

Student will be able to

Understand conception of parser

Use Yacc generators

SW & HW Packages Required:

Working PC

64 bit Fedora PC

GCC Compiler

Lex Compiler

Yacc Compiler

Editor (Text)

14 Learning techniques by virtue of advance programming skill set and FOSS tools.

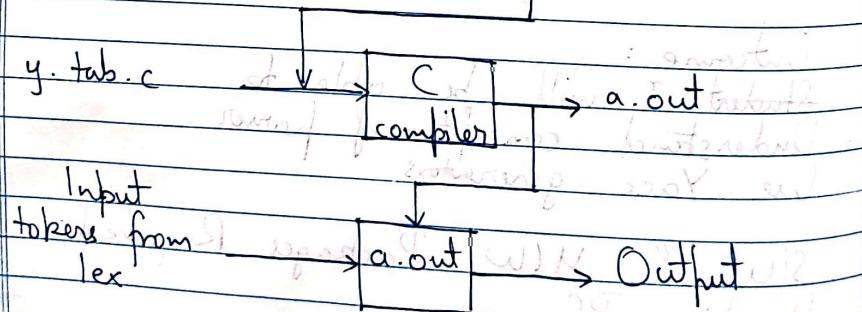
Theory

2.8 grammars

Yacc is a parser generator which is used with lex (lexical analyser generator) in UNIX operating system. Both lex & yacc are system tools for compiler construction.

Yacc assists in parsing phase. It creates a C program for parser. It can be used as a command on Linux.

Yacc is Functioning like this. It takes yacc specification as input and outputs a.out file.



Parsing

Parsing is the activity of checking whether a string of symbols is in the language of some grammar where the string is usually the stream of tokens produced by the lexical analyser.

If the string is in the grammar, we want a parse tree. If not, we want some kind of an error message.

There are two types of parsers:

i] Top - down Parser

A top down parser attempts to construct a tree from the root. It applies production forwards to expand non-terminals into strings of symbols.

ii] Bottom - up Parser

A bottom up parser builds the tree starting with trees. It uses the production in reverse to identify strings of symbols that can be grouped together.

In both cases the construction of derivation is directed by scanning the input sequence from left to right one symbol at a time.

YACC is an LALR(1) Parser generator
(Lookahead, Left-to-Right, is a Rightmost derivation producer with 1 lookahead token)

Input of YACC consists of the rules of grammar

Output of YACC is a C-program
(y.tab.c)

YACC specifications for union type

1. % start line

↳ The whole input should match lines start

2. % union

↳ Lists all possible types for the values associated with each part of grammar using field name from '% union' declaration.

3. % type

↳ gives an individual type for values associated with each part of grammar using field name from '% union' declaration

4. % token

↳ declare each grammar rule used by YACC that is recognised by LEX & give type of value

Page: 6
Date: 11/1

Test Cases

Input	Expected Output	Actual Output	Result
i) Apoorv is in PICT	Apoorv : Noun is : Verb in : Preposition PICT : Noun Simple Sentence	Apoorv : Noun is : Verb in : Preposition PICT : Noun Simple Sentence	Success
ii) Apoorv is boy and he is student	Apoorv : Noun is : Verb boy : Noun and : Conjunction he : Pronoun is : Verb student : Noun Compound Sentence	Apoorv : Noun is : Verb boy : Noun and : Conjunction he : Pronoun is : Verb student : Noun Compound Sentence	Success

Conclusion: Successfully Implemented Syntax Analysis Phase of compiler to recognise Simple & Compound sentences given in input file using YACC specifications