

Compiler Design and Construction

Course Code : 540203

Incourse suggestion :Chapter:1,2 and 3

1.Introduction to compiler: Compiler, Analysis of the source Program, the phases of compiler, of the compiler, compiler construction tools.

2.A simple one pass compiler: syntax definition, CFG, parse tree, ambiguity, associativity of operators, lexical analysis.

3.Lexical analysis: the role of the lexical analyzer, input buffering, specification tokens, finite automaton, Thompson's construction, conversion of regular expression to DFA

Chapter-1 (Introduction to Compiling)

1. What are the phases of compiler?Describe different phases of compiler with example
2. Define compiler.Illustrate the representation of the following statement after each phase of compiler: position=initial+rate*60.
3. What do you mean by translator software?Differentiate between compiler and interpreter.
4. What is preprocessor?Describe the functions of preprocessor.
5. Define pass,cross-compiler and preprocessor.
6. Define symbol table?What are the functions of symbol table.
7. Describe language processing system.
8. What are the types of errors found in different phase of compiler.

Chapter-2(One pass compiler)

1. Describe the plan of an error detector/corrector with figure
2. Write about Error Recovery Strategies in Compiler Design
3. Define error.Describe the source of errors
4. What are the properties of good error diagnostics?
5. What do you mean by context free grammar ?What are the components of grammar?
6. Define parse tree and describe its properties.
7. What is ambiguous grammar?Explain with example.
8. Describe the associativity of operators with example.
9. Translate the following infix expression into its equivalent prefix expression or postfix expression:
(i) $(A+B*D)/(E-F)+G$
(ii) $A*(B+D)/E-F*(G+H/K)$
10. Consider the following grammar for arithmetic expression involving +,-,*,/ and
 $E \rightarrow E+E \mid E-E \mid E * E \mid E / E \mid E \uparrow E \mid (E) \mid -E \mid id$

Is it an ambiguous grammar? Justify your answer.

11. Derive a parse tree for $9-5+2$ according to following rules or productions.

list \rightarrow list+digit(1)

list \rightarrow list-digit(2)

list \rightarrow digit(3)

digit \rightarrow 0|1| 2| 3| 4 |5 |6 |7| 8 |9.....(4)

12. Consider the context free grammar-

$S \rightarrow SS+|SS^*|a$

(i) Show how the string $aa+a^*$ can be generated by this grammar.

(ii) Construct a parse tree for this string.

13. Consider the following Context free grammar(CFG):

stmt \rightarrow expr;|expr;stmt

expr \rightarrow expr+term|term

term \rightarrow term*factor|factor

factor \rightarrow number|(expr)

number \rightarrow 0|1|2|.....|9|

(i) Show how the string $1+2*(3+4)+5$; can be generated by this grammar.

(ii) Construct a parse tree for this string.

14. Consider the following grammar:

$S \rightarrow iCtS$

$S \rightarrow iCtSeS$

$S \rightarrow a$

$S \rightarrow b$

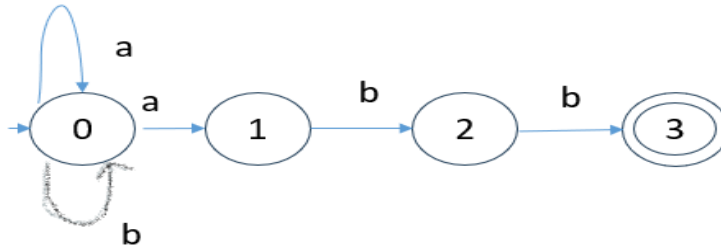
Here I, t and e stand for if, then and else, C and S for "conditional" and

"statement". Construct a parse tree for the sentence w-ibtibtaea using left most derivation and right most derivation.

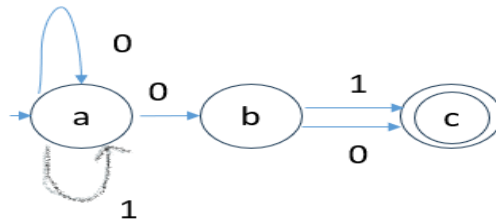
Chapter-3(Lexical Analysis)

1. Define token, pattern and lexeme with example.
2. Discuss the relationship among input, lexical analyzer and a parser.
3. What are the reasons for separating the analysis phase of compiling into lexical analysis and parsing?

4. Define Finite Automata. What are the difference between DFA and NFA.
5. Describe DFA and NFA.
6. Design an NFA that accept language $aa^*|bb^*$.
7. Construct a DFA from the following NFA:



8. Construct a DFA from the following NFA:



9. Construct a DFA and NFA for the regular expression $(a|b)^*abb$
10. The NFA accepting the language $(a|b)^*abb$; Construct the DFA using the subset construction rule./From the following NFA accepting the language $(a|b)^*abb$; Construct the DFA using the subset construction rule.

