Group-A

Module I

1. What are the functionalities of a Compiler?
2. What is the purpose of Lexical Analyzer?
3. What is the purpose of Syntax Analyzer?
4. What is the purpose of Semantic Analyzer?
5. What are the different components of front end compilation process?
6. Why front end compilation process is machine independent?
7. What is the purpose of Intermediate code generator?
8. What is the purpose of Code optimizer?
9. What is the purpose of Code generator?
10. What do you mean by Parse tree?
11. What is ambiguous parse tree?
12. How do we remove ambiguity from parse tree?
13. What is Symbol Table and explain its purpose?
14. What is 'Three address code'?
15. What are the different representations of three address code?
16. When can the source program and target program be said equivalent?

Module II

17. What are the front end phases of compilation process?
18. Design a DFA that accepts all strings over $\Sigma = \{a\}$?
19. Write down the regular expression for the following language;
    Set of all strings containing at least 2 a's over $\Sigma = \{a, b\}$
20. Write down the regular expression for the following language;
    Set of all strings containing exactly 2 a's over $\Sigma = \{a, b\}$
21. Write down the regular expression for the following language;
    Set of all even length strings over $\Sigma = \{a, b\}$
22. Write down the regular expression for the following language;
    Set of all strings whose lengths are divisible by 3, over $\Sigma = \{a, b\}$
23. Write down the regular expression for the following language;
    Set of all strings that begin and end with different symbols, over $\Sigma = \{a, b\}$
24. Write down the regular expression for the following language;
    Set of all strings that ends with 'aab' over $\Sigma = \{a, b\}$
25. Write down the regular expression for the following language;
    Set of all strings that begin with 'bab' over $\Sigma = \{a, b\}$
26. Write down the regular expression for the following language;
    Set of all strings that ends with 'aba' over $\Sigma = \{a, b\}$
27. Write down the regular expression for the following language;
    Set of all strings that begin with 'aba' over $\Sigma = \{a, b\}$
28. Draw a DFA that accepts the following language;
    Set of all strings that contains even number a's over $\Sigma = \{a\}$
29. Draw a DFA that accepts the following language;

Set of all strings that contains odd number a’s over $\Sigma = \{a\}$

30. Consider the following grammar; $S \rightarrow S\,S + \mid S\,S * \mid a$. Show that the string ‘$aa+a*$’ can be generated by this grammar?
31. Consider the following grammar; $S \rightarrow S\,S + \mid S\,S * \mid a$. Show that the string ‘$aa*a+$’ can be generated by this grammar?
32. Check if the following string ‘1010110’ can be obtained from the following regular expression

    (0 + 1)*011(1 + 0)
33. Check if the following string ‘101010’ can be obtained from the following regular expression

    (0 + 1)*011(1 + 0)*
34. Check if the following string ‘1101011’ can be obtained from the following regular expression

    0*101*0*
35. Check if the following string ‘011’ can be obtained from the following regular expression

    (0 + 1)*011(1 + 0)*
36. Consider the following grammar; $S \rightarrow S\,S + \mid S\,S * \mid a$. Show that the string ‘$aa+$’ can be generated by this grammar?
37. Consider the following grammar; $S \rightarrow S\,S + \mid S\,S * \mid a$. Show that the string ‘$aa*$’ can be generated by this grammar?
38. Consider the following grammar; $S \rightarrow S(S)S \mid a$. Show that the string ‘$a(a)a(a)a$’ can be generated by this grammar?
39. Consider the following grammar; $E \rightarrow E + E \mid E * E \mid - E \mid (E) \mid id$. Show that the string ‘$id * id+id$’ can be generated by this grammar?
40. Consider the following grammar; $E \rightarrow E + E \mid E * E \mid - E \mid (E) \mid id$. Show that the string ‘$(id+id)$’ can be generated by this grammar?
41. Consider the following grammar; $E \rightarrow E + E \mid E * E \mid - E \mid (E) \mid id$. Show that the string ‘$(id* (-id))$’ can be generated by this grammar?
42. Check if the following string ‘100100’ can be obtained from the following regular expression

    (0 + 1)*011(1 + 0)*
43. Check if the following string ‘10010’ can be obtained from the following regular expression

    (0 + 1)*011(1 + 0)*
44. Check if the following string ‘10010’ can be obtained from the following regular expression

    01(0 + 1)*011
45. Check if the following string ‘011010011’ can be obtained from the following regular expression

    01(0 + 1)*011
46. Consider the following grammar; $S \rightarrow S\,S + \mid S\,S * \mid a$. Show the rightmost derivation of the string ‘$aa+$’?
47. Consider the following grammar; $S \rightarrow S\,S + \mid S\,S * \mid a$. Show the rightmost derivation of the string ‘$aa*$’?

48. Consider the following grammar; $S \rightarrow S(S)S \mid a$. Show the rightmost derivation of the string ‘$a(a)a(a)a$’?
49. Consider the following grammar; $E \rightarrow E + E \mid E * E \mid - E \mid (E) \mid id$. Show the rightmost derivation of the string ‘$id * id + id$’?
50. What do you mean by a language?
51. What do you mean by an alphabet?
52. What do you mean by a symbol?
53. What do you mean by acceptance of a language by a DFA?
54. What is LEX?

Module III

55. What do you mean by syntax tree?
56. What is left recursion?
57. Why left recursion is considered to be a problem in parsing process?
58. Remove the left recursion from the following grammar: $A \rightarrow A\alpha \mid \beta$
59. What do you mean by ambiguity of a grammar?
60. What do you mean by Handle?
61. What do the L’s in $LL(1)$ grammar signify?
62. If ‘X’ is a terminal then what is FIRST(X)?
63. Define GOTO function associated with Shift- Reduce parser?
64. Define LR grammar?
65. What do the L and R in $LR(1)$ grammar signify?
66. Define Augmented Grammar?
67. What is Closure of an Item set?
68. Define FIRST sets associated with parsing?
69. Define FOLLOW sets associated with parsing?
70. What do you mean by top-down parsing?
71. What do you mean by predictive parsing?
72. What is Recursive Descent parsing?
73. What do you mean by handle pruning?
74. What do you mean by bottom-up parsing?
75. Explain the Shift & Reduce operation for bottom-up parsing?
76. How do you check if a grammar is LL(1) or not?
77. What is Shift-Reduce parser?
78. What is the utility of ‘Item’ sets in Bottom-up parsing?
79. What is dangling-else problem?
80. What is YACC?

Module IV

81. What is a Directed Acyclic Graph?
82. Draw the DAG of the following expression: $i = i + 10$
83. Draw the DAG of the following expression: $a = a + b$
84. Draw the DAG of the following expression: $x = x + (x + y)$
85. Draw the DAG of the following expression: $x > (y - x) + y$

86. Write down the three address code of the following expression: $i = i + 10$
87. Write down the three address code of the following expression: $a = a + b$
88. Write down the three address code of the following expression: $x = x + (x + y)$
89. Write down the three address code of the following expression: $x > (y - x) + y$
90. How DAG is used to find common sub-expression?
91. What do you mean by Three Address Code?
92. What is copy instruction form of three address code?
93. How does a conditional jump instruction written using three address code?
94. What do you mean by Quadruples?
95. What do you mean by Triples?
96. What do you mean by Indirect Triples?
97. What do you mean by Code optimization?
98. What is Peephole optimization?
99. What is an Activation Record?
100. What are the different components of Activation record?

Group-B

Module I

101. Explain the working principle of Lexical Analyzer with a suitable example?
102. Explain the working principle of Syntax Analyzer with a suitable example?
103. Explain the working principle of Semantic Analyzer with a suitable example?
104. Explain the working principle of Intermediate code generator with a suitable example?
105. Explain the working principle of Code optimizer with a suitable example?
106. Explain the working principle of Code generator with a suitable example?
107. Explain the following terms with suitable examples: symbol, alphabet, string, language.
108. What do you mean by Symbol Table? Consider the expression ‘a = b + c;’ and show the symbol table entries after lexical analysis?
109. Consider the expression ‘a = bc;’ At which phase of compilation process the error will be encountered? Explain why it is not possible to detect the error in previous phases?

Module II

110. Write a short note on Syntax tree.
111. What do you mean by ambiguity of a grammar? Explain with a suitable example?
112. Draw the parse tree of the expression 9-5-2 considering both left and right associativity?
113. Consider the following grammar;

$S \rightarrow S\,S + \mid S\,S\,* \mid a$

a. Show that the string ‘*aa+a**’ can be generated by this grammar ?
b. Construct the parse tree ?

114. Consider the following grammar;

$S \rightarrow S + S \mid S * S \mid 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Construct the parse tree of ‘$9 – 5 + 2$’ using this grammar

115. What do you mean by L-values and R-values? Explain with suitable example?
116. Write short note on ‘Lex’?
117. Convert the following regular expression to a DFA;

$(a + b)*abb(a + b)*$

118. Convert the following NFA to an equivalent DFA;

State	*a*	*b*
q_0	q_0, q_1	q_0
q_1	φ	q_2
q_2	φ	φ

119. Prove that the following grammar is ambiguous by explaining it using a suitable example;

$E \rightarrow E + E \mid E * E \mid - E \mid (E) \mid id$

120. Show that the string ‘id + id * id’ has two distinct leftmost derivation for the following grammar;

$E \rightarrow E + E \mid E * E \mid - E \mid (E) \mid id$

121. Consider the following grammar;

$S \rightarrow S\,S + \mid S\,S\,* \mid a$

a. Show the leftmost derivation of string ‘*aa+a**’?
b. Show the rightmost derivation of string ‘*aa+a**’?
c. Is the grammar ambiguous or unambiguous?

122. (i) Write down the regular expression for the following language;

Set of all strings whose lengths are divisible by 3, over $\Sigma = \{a, b\}$

(ii) Write down the regular expression for the following language;

Set of all strings that begin and end with different symbols, over $\Sigma = \{a, b\}$

123. (i) Write down the regular expression for the following language;

Set of all strings that ends with 'aab' over $\Sigma = \{a, b\}$

(ii) Write down the regular expression for the following language;

Set of all strings that begin with 'bab' over $\Sigma = \{a, b\}$

124. Write a context free grammar that generates all palindromes over $\Sigma = \{a, b\}$

Module III

125. What do you mean by left recursion problem? Propose a solution to remove left recursions?

126. What do you mean by left recursion problem? Remove the left recursion from the following grammar

$A \rightarrow A\alpha \mid \beta$

127. Remove the left recursion from the following grammar

$E \rightarrow E + T \mid T$
$T \rightarrow T * F \mid F$
$F \rightarrow (E) \mid id$

128. Define left factoring? How will you eliminate left factor?

129. Write an algorithm to compute FIRST and FOLLOW sets associated with parsing?

130. Write an algorithm to produce the predictive parsing table while the grammar and the FIRST and FOLLOW sets of the grammar symbols are given.

131. Remove the left recursion from the following grammar;

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

132. Find the FIRST and FOLLOW of every grammar symbol of the following grammar;

$$E \rightarrow TE'$$
$$E' \rightarrow +TE'|\varepsilon$$
$$T \rightarrow FT'$$
$$T' \rightarrow *FT'|\varepsilon$$
$$F \rightarrow (E) \mid id$$

133. Parse the string $id + id * id$ by top down parsing method using the following grammar;

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

134. Find the FIRST and FOLLOW of every grammar symbol of the following grammar;

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

135. Parse the string $acdb$ by top down parsing method using the following grammar;

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

136. Parse the string ab by top down parsing method using the following grammar;

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

137. Parse the string $acdb$ by top down parsing method using the following grammar;

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

138. Parse the string acb by top down parsing method using the following grammar;

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

139. Parse the string adb by top down parsing method using the following grammar;

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

140. Find the FIRST and FOLLOW of every grammar symbol of the following grammar;

$$S \rightarrow AaAb \mid BbBa$$
$$A \rightarrow \epsilon$$
$$B \rightarrow \epsilon$$

141. Parse the string $aaab$ by top down parsing method using the following grammar;

$$S \to AaAb \mid BbBa$$
$$A \to \epsilon|a$$
$$B \to \epsilon\ |b$$

142. Parse the string $bbba$ by bottom up parsing method using the following grammar;

$$S \to AaAb \mid BbBa$$
$$A \to \epsilon|a$$
$$B \to \epsilon\ |b$$

143. Parse the string ab by top down parsing method using the following grammar;

$$S \to AaAb \mid BbBa$$
$$A \to \epsilon|a$$
$$B \to \epsilon\ |b$$

144. Parse the string aab by bottom up parsing method using the following grammar;

$$S \to AaAb \mid BbBa$$
$$A \to \epsilon|a$$
$$B \to \epsilon\ |b$$

145. Parse the string bba by top down parsing method using the following grammar;

$$S \to AaAb \mid BbBa$$
$$A \to \epsilon|a$$
$$B \to \epsilon\ |b$$

146. Parse the string ba by bottom up parsing method using the following grammar;

$$S \to AaAb \mid BbBa$$
$$A \to \epsilon|a$$
$$B \to \epsilon\ |b$$

147. Define $LL(1)$ grammar? How to decide if a given grammar is $LL(1)$ or not?

148. Consider the grammar,

$$S \to (L)|A$$
$$L \to L, S \mid S$$

a) What are the terminals, on terminals and start symbol

b) find parse tree for (i)(a,a) (ii) (a,(a,a) (iii)(a,((a,a),(a,a)))

149. Explain the working principle of LR parsers

150. Explain top down parsing using a suitable example

151. Explain bottom parsing using a suitable example

152. Parse the string $(id + id)$ by shift-reduce parsing method using the following grammar;

$$E \to E + T \mid T$$
$$T \to T * F \mid F$$
$$F \to (E) \mid id$$

153. Parse the string $id * (id + id)$ by shift-reduce parsing method using the following grammar;

$$\begin{gathered} E \rightarrow E + T \mid T \\ T \rightarrow T * F \mid F \\ F \rightarrow (\, E \,) \mid id \end{gathered}$$

154. Parse the string $(id * id) + id$ by shift-reduce parsing method using the following grammar;

$$\begin{gathered} E \rightarrow E + T \mid T \\ T \rightarrow T * F \mid F \\ F \rightarrow (\, E \,) \mid id \end{gathered}$$

155. Define LL(1) grammar with examples
156. Explain recursive descent parsing using a suitable example
157. Explain predictive parsing using a suitable example
158. Parse the string $id * id$ by shift-reduce parsing method indicating the handle at each step. Draw the bottom-up parse tree.

$$\begin{gathered} E \rightarrow E + T \mid T \\ T \rightarrow T * F \mid F \\ F \rightarrow (\, E \,) \mid id \end{gathered}$$

159. Parse the string $id + id * id$ by shift-reduce parsing method using the following grammar;

$$\begin{gathered} E \rightarrow E + T \mid T \\ T \rightarrow T * F \mid F \\ F \rightarrow (\, E \,) \mid id \end{gathered}$$

160. What do you mean by handle? What is handle pruning?
161. Write an algorithm to compute the closure set of an Item associated with Shift-Reduce parser.
162. Write a short note on YACC.

Module IV

163. What do you mean by Directed Acyclic Graph? Explain using a suitable example?
164. Draw the DAG of the following expression; $a + a * (b - c) + (b - c) * d$
165. Draw the DAG of the following expression;

$$((x + y) - ((x + y) * (x - y))) + ((x + y) * (x - y))$$

166. Draw the DAG of the following expression;

$$((x + y) - ((x + y) * (x - y)))$$

167. Draw the DAG of the following expression;

$$x = (x - ((x + y) * (x - y))) + ((x + y) * (x - y))$$

168. Draw the DAG of the following expression;

$$((a - b) * ((a * b) * (a - b)))$$

169. Draw the DAG of the following expression; $x + x * (y + z) - (y + z) - w$

170. What do you mean by Three-Address code? Find the three-address code of the following expression;

$$a + a * (b - c) + (b - c) * d$$

171. What do you mean by Three-Address code? Find the three-address code of the following expression;

$$a = b * - c + b * - c$$

172. Write a short note on Three-address code.

173. What do you mean by Quadruple? Find the quadruple representation of the following three-address code;

$$\begin{aligned} t_1 &= minus\ c \\ t_2 &= b * t_1 \\ t_3 &= minus\ c \\ t_4 &= b * t_3 \\ t_5 &= t_3 + t_4 \\ a &= t_5 \end{aligned}$$

174. What do you mean by Triples? Find the Triples representation of the following three-address code;

$$\begin{aligned} t_1 &= minus\ c \\ t_2 &= b * t_1 \\ t_3 &= minus\ c \\ t_4 &= b * t_3 \\ t_5 &= t_3 + t_4 \\ a &= t_5 \end{aligned}$$

175. What do you mean by Indirect Triples? Find the Indirect Triples representation of the following three-address code;

$$\begin{aligned} t_1 &= minus\ c \\ t_2 &= b * t_1 \\ t_3 &= minus\ c \\ t_4 &= b * t_3 \\ t_5 &= t_3 + t_4 \\ a &= t_5 \end{aligned}$$

176. What are the common three-address instruction forms? Explain
177. Write short note on Quadruple.
178. Write short note on triples.
179. Write short note on indirect triples.
180. Define a DAG? Explain how is it used to identify common subexpression?
181. Write short note on Activation Record.

182. Write short note on Peephole optimization.
183. How Peephole optimization is used to eliminate unreachable code? Explain?
184. How Peephole optimization is used to optimize flow-of-control? Explain?
185. What is Activation Record? Explain its different components?

Group-C

Module I

186. Explain the different phases of compilation with a suitable example?
187. Explain how the following expression is translated via different phases of compilation using a suitable diagram;

position = initial + rate * 60

188. Write short note on ***any three*** of the following:
  i. Three-address code
  ii. Syntax tree
  iii. Symbol table
  iv. Lexical Analysis
  v. Back end compilation

Module II

189. What do you mean by Symbol Table? Consider the expression 'a = bc;' and show the symbol table entries after lexical analysis? At which phase of compilation process the error will be encountered? Explain why it is not possible to detect the error in previous phases? **4+5+2+4**
190. What do you mean by ambiguity of a grammar? Explain with a suitable example? Consider the following grammar; Show that the string '$aa + a *$' can be generated by this grammar? Construct the parse tree? $S \to S\,S + \mid S\,S * \mid a$ **2+3+5+5**

191. What do you mean by ambiguity of a grammar? Explain with a suitable example? Consider the following grammar; Show that the string '$((a))$' can be generated by this grammar? Construct the parse tree? $S \to (S) \mid S + S \mid a$ **2+3+5+5**

192. Convert the following regular expression to a DFA;

$$(a + b)^* \, abb \, (a + b)^*$$

Convert the following NFA to an equivalent DFA;

State	**a**	**b**
→ q_0	q_0, q_1	q_0

q_1	φ	q_2
* q_2	φ	φ

6+9

193. Convert the following regular expression to a DFA;

$$a^* \, abb \, (a + b)^*$$

Convert the following NFA to an equivalent DFA;

State	***a***	***b***
→* q_0	q_0	q_1
q_1	q_1	q_0, q_1

6+9

194. Convert the following regular expression to a DFA;

$$(a + b)^* \, abb \, (a + b)^*$$

Convert the following NFA to an equivalent DFA;

State	***a***	***b***
→ q_0	q_0, q_1	q_2
q_1	q_0	q_1
* q_2	φ	q_0, q_1

6+9

195. Convert the following regular expression to a DFA;

$$(a + b)^* \, baa \, b^*$$

Convert the following NFA to an equivalent DFA;

State	***a***	***b***
→ q_0	q_0, q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
*q_3	φ	q_2

6+9

196. Convert the following regular expression to a DFA;

$$a^* \, abb \, (a + b)^*$$

Convert the following NFA to an equivalent DFA;

State	***a***	***b***
→ q_0	q_0, q_1	q_1
* q_1	φ	q_0, q_1

6+9

197. Construct an equivalent NFA of the following regular expression; (0+1)*(00+11)(0+1)*. Convert that NFA to its equivalent DFA? **7+8**

198. Construct an equivalent NFA of the following regular expression; 0*(00+1)(0+1)*. Convert that NFA to its equivalent DFA? **7+8**

199. Construct an equivalent NFA of the following regular expression; 0*(00+11)1*. Convert that NFA to its equivalent DFA? **7+8**

200. Construct an equivalent NFA of the following regular expression; (0+1)*(0+1)1*. Convert that NFA to its equivalent DFA? **7+8**

201. Show that the string '$id + id * id$' has two distinct leftmost derivation for the following grammar;

$$E \rightarrow E + E \mid E * E \mid - E \mid (E) \mid id$$

Consider the following grammar;

$$S \rightarrow S S + \mid S S * \mid a$$

vi. Show the leftmost derivation of string '$aa + a *$'?

vii. Show the rightmost derivation of string '$aa + a *$'?

viii. Is the grammar ambiguous or unambiguous? **8+7**

202. Write down the regular expression for the following language;

(i) Set of all strings whose lengths are divisible by 3, over $\Sigma = \{a, b\}$

(ii) Set of all strings that begin and end with different symbols, over $\Sigma = \{a, b\}$

(iii) Set of all strings that ends with 'aab' over $\Sigma = \{a, b\}$

(iv) Set of all strings that begin with 'bab' over $\Sigma = \{a, b\}$

(v) Set of all even length strings over $\Sigma = \{a, b\}$ **3+3+3+3+3**

203. Write down the regular expression for the following language;

(i) Set of all strings containing at least 2 a's over $\Sigma = \{a, b\}$

(ii) Set of all strings containing exactly 2 a's over $\Sigma = \{a, b\}$

(iii) Set of all even length strings over $\Sigma = \{a, b\}$

(iv) Set of all strings whose lengths are divisible by 3, over $\Sigma = \{a, b\}$

(v) Set of all strings that begin and end with different symbols, over $\Sigma = \{a, b\}$ **3+3+3+3+3**

204. Show that the string '$id + id * id$' has two distinct leftmost derivation for the following grammar;

$$E \rightarrow E + E \mid E * E \mid - E \mid (E) \mid id$$

Consider the following grammar;

$$S \rightarrow (S) \mid S + S \mid a$$

ix. Show the leftmost derivation of string ‘$(a) + (a)$’?

x. Show the rightmost derivation of string ‘$(a) + (a)$’?

xi. Draw the parse tree? **8+7**

205. (i) Convert the following NFA to an equivalent DFA;

State	**a**	**b**
→q_0	{ q_0, q_1}	{ q_2}
q_1	{ q_0}	{ q_1}
*q_2	φ	{ q_0, q_1}

(ii) Find the regular expressions of the following languages;

a) Set of all strings over {a, b} such that all the strings begin and end with different symbols.

b) Set of all strings over {a, b} such that all the strings begin and end with same symbol.

9+6

206. (i) Convert the following NFA to an equivalent DFA;

State	**a**	**b**
→q_0	{ q_0, q_1}	{ q_2}
q_1	{ q_0}	{ q_1}
*q_2	φ	{ q_0, q_1}

(ii) Find the regular expressions of the following languages;

a) Set of all strings containing at least 2 a’s over $\Sigma = \{a, b\}$

b) Set of all strings containing exactly 2 a’s over $\Sigma = \{a, b\}$

c) Set of all even length strings over $\Sigma = \{a, b\}$ **9+6**

207. (i) Convert the following NFA to an equivalent DFA;

State	***a***	***b***
→ q_0	q_0, q_1	q_0
q_1	φ	q_2
* q_2	φ	φ

(ii) Find the regular expressions of the following languages;

a) Set of all strings containing at least 2 a’s over $\Sigma = \{a, b\}$

b) Set of all strings containing exactly 2 a's over $\Sigma = \{a, b\}$

c) Set of all even length strings over $\Sigma = \{a, b\}$ **9+6**

Module III

208. Remove the left recursion from the following grammar

$E \rightarrow E + T \mid T$
$T \rightarrow T * F \mid F$
$F \rightarrow (E) \mid id$

Prove that the following grammar is ambiguous by explaining it using a suitable example;

$E \rightarrow E + E \mid E * E \mid - E \mid (E) \mid id$ **7+8**

209. What do you mean by left recursion problem? Propose a solution to remove left recursions. Remove the left recursion from the following grammar; **5+4+6**

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

210. Remove the left recursion from the following grammar. After that find the FIRST and FOLLOW of every grammar symbol **7+8**

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

211. What is left factoring? What do you mean by 'Dangling-else' problem? Represent it in form of a grammar? Left factor the grammar? **4+3+4+4**

212. What is top down parsing? Define bottom up parsing? Consider the following grammar and draw the top down parsing tree of the of the following string $id * id + id$;

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid id$$

3+3+9

213. Write an algorithm to compute FIRST and FOLLOW sets associated with parsing? Find the FIRST and FOLLOW of every grammar symbol of the following grammar. **7+8**

$$E \rightarrow TE'$$
$$E' \rightarrow +TE'|\varepsilon$$
$$T \rightarrow FT'$$
$$T' \rightarrow *FT'|\varepsilon$$
$$F \rightarrow (E) \mid id$$

214. Write an algorithm to compute FIRST and FOLLOW sets associated with parsing? Find the FIRST and FOLLOW of every grammar symbol of the following grammar. **7+8**

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

215. Find the FIRST and FOLLOW of every grammar symbol of the following grammar. Check if the grammar is LL(1) or not? **7+8**

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

216. Construct the predictive parsing table to check if the grammar is LL(1) or not? Parse the string $acdb$ by using the parsing table; **7+8**

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

217. Construct the predictive parsing table to check if the grammar is LL(1) or not? Parse the string acb by using the parsing table; **7+8**

$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

218. Construct the predictive parsing table to check if the grammar is LL(1) or not? Parse the string ‘ba’ and ‘ab’ by using the parsing table; **7+8**

$$S \rightarrow AaAb \mid BbBa$$

$$A \rightarrow \epsilon$$
$$B \rightarrow \epsilon$$

219. Prove that the following grammar is LL(1)? Parse the string ‘ba’ by using the parsing table; **8+7**

$$S \rightarrow AaAb \mid BbBa$$
$$A \rightarrow \epsilon$$
$$B \rightarrow \epsilon$$

220. Prove that the following grammar is LL(1)? Parse the string ‘aa’ by using the parsing table; **8+7**

$$S \rightarrow SA \mid A$$
$$A \rightarrow a$$

221. Construct the predictive parsing table to check if the grammar is $LL(1)$ or not? Parse the string $id + id * id$ by using the parsing table; **7+8**

$$E \rightarrow TE'$$
$$E' \rightarrow +TE'|\varepsilon$$
$$T \rightarrow FT'$$
$$T' \rightarrow *FT'|\varepsilon$$
$$F \rightarrow (E) \mid id$$

222. Find the FIRST and FOLLOW of every grammar symbol of the following grammar. Construct the predictive parsing table to check if the grammar is $LL(1)$ or not? Parse the string ‘ab’ using the parsing table. **5+5+5**

$$S \rightarrow AaAb \mid BbBa$$
$$A \rightarrow \epsilon$$
$$B \rightarrow \epsilon$$

223. What do you mean by ‘Dangling-else’ problem? Represent it in form of a grammar? Left factor the grammar? Construct the predictive parsing table to check if the grammar is $LL(1)$ or not? **2+3+3+7**

224. What is left factoring? Left factor the following grammar? Construct the predictive parsing table to check if the grammar is $LL(1)$ or not?

$$S \rightarrow iCtS \mid iCtSeS \mid a$$
$$C \rightarrow b$$

225. Find the FIRST and FOLLOW of every grammar symbol of the following grammar. Construct the predictive parsing table to check if the grammar is $LL(1)$ or not? Parse the string ‘ba’ using the parsing table. **5+5+5**

$$S \rightarrow AaAb \mid BbBa$$
$$A \rightarrow \epsilon$$
$$B \rightarrow \epsilon$$

226. What do you mean by an 'Item' in shift-reduce parsing? Define GOTO function? Explain how the LR(0) automaton is constructed using GOTO function? Show the bottom up parse tree of the string $acdb$ by using the following grammar; **3+3+4+5**

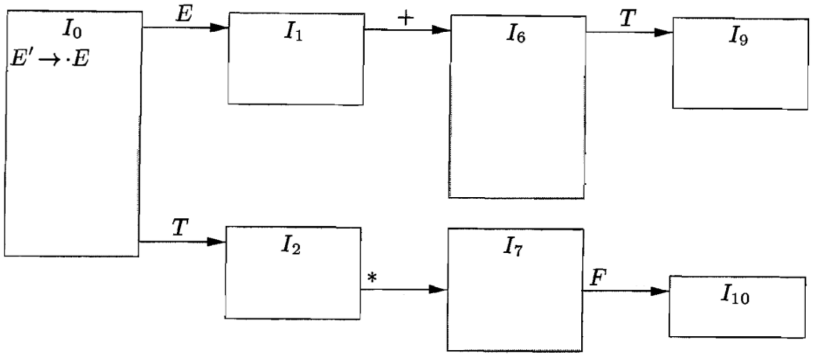
$$S \rightarrow aABb$$
$$A \rightarrow c \mid \epsilon$$
$$B \rightarrow d \mid \epsilon$$

227. Parse the string $id * id + id$ by shift-reduce parsing method indicating the input string, stack entry and action taken at each step. Draw the bottom-up parse tree. **10+5**

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

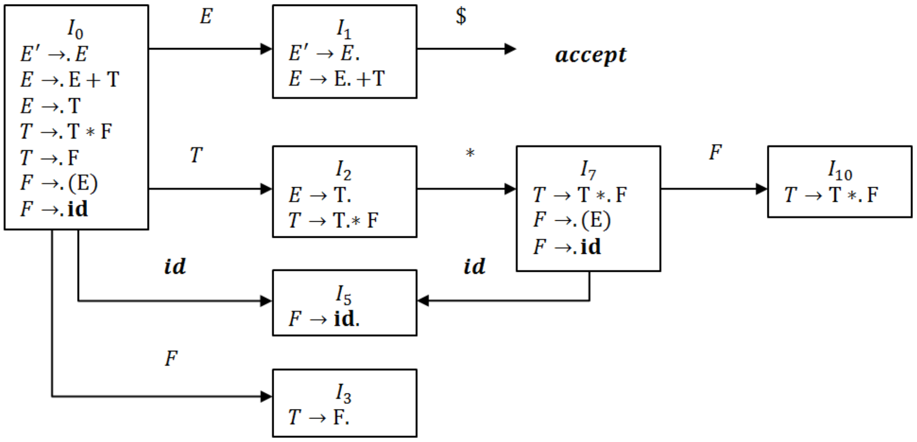
228. Write an algorithm to compute the closure set of an Item associated with Shift-Reduce parser. Consider the following grammar and complete the closure sets in the following SLR automaton.

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$



5+10

229. What is bottom up parsing? Consider the following LR(0) automaton and parse the string $id * id$. Show Stack, grammar symbols corresponding to the states, input and action entries.



3+12

230. What do you mean by handle? What is handle pruning? Parse the string $id * id$ by shift-reduce parsing method using the following grammar indicating the handle at each step;

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

3+4+8

231. Parse the string $(id) * id$ by shift-reduce parsing method indicating the input string, stack entry and action taken at each step. Draw the bottom-up parse tree. **10+5**

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

232. What do you mean by left recursion problem? Propose a solution to remove left recursions. Remove the left recursion from the following grammar; **5+4+6**

$$A \rightarrow Ac \mid Aad \mid bd \mid \epsilon$$

233. What is top down parsing? Define bottom up parsing? Consider the following grammar and draw the bottom up parsing tree of the of the following string $id * id + id$;

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$

3+3+9

234. What is right most derivation? Explain using a suitable example? Consider the following grammar and show the leftmost and rightmost derivation of the of the following string $id * id + id$;

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid id$$

6+9

Module IV

235. What do you mean by Directed Acyclic Graph? Draw the DAG of the following expression: $a + a * (b - c) + (b - c) * d$ Write down the three-address code of the same expression. **4+5+6**

236. What do you mean by Three-Address code? Find the three-address code of the following expression;

$$a + a * (b - c) + (b - c) * d$$

Draw the DAG of the following expression;

$$((x + y) - ((x + y) * (x - y))) + ((x + y) * (x - y))$$ **8+7**

237. Write a short note on Three-address code. Find the three-address code of the following expression. Find the quadruple representation of the three address code. **5+5+5**

$$a = b * -c + b * -c$$

238. What do you mean by Three-address code? Find the three-address code of the following expression. Find the triples representation of the three address code. **5+5+5**

$$a = b * -c + b * -c$$

239. Find the three-address code of the following expression. What do you mean by indirect triples? Find the indirect triples representation of the three address code. **5+5+5**

$$a = b * -c + b * -c$$

240. Find the quadruples, triples and indirect triples representation of the following three-address code; **5+5+5**

$$\begin{aligned} t_1 &= minus\ c \\ t_2 &= b * t_1 \\ t_3 &= minus\ c \\ t_4 &= b * t_3 \\ t_5 &= t_3 + t_4 \\ a &= t_5 \end{aligned}$$

241. Find the three-address code of the following expression. Find the quadruple and triples representation of the three address code. **5+5+5**

$$a = b * -c + b * -c$$

242. Write short note on ***any three*** of the following:

i. Lex
ii. YACC
iii. Syntax tree
iv. Three-address code
v. Peephole optimization

243. Find the three-address code of the following expression. Find the triples and indirect triples representation of the three address code. **5+5+5**

$$a = b * -c + b * -c$$

244. Draw the DAG of the following expression. Find the three-address code of the following expression. Find the quadruples representation of the three address code. **5+5+5**

$$a = b * -c + b * -c$$

245. Write short note on ***any three*** of the following:
i. Activation record
ii. Quadruples
iii. DAG
iv. Top-down parsing
v. Lex

246. Write short note on ***any three*** of the following:
i. Symbol table
ii. Left Recursion
iii. YACC
iv. Triples
v. DAG

247. Write short note on ***any three*** of the following:
i. Activation record
ii. Left Recursion
iii. Indirect Triples
iv. Bottom up parsing
v. Lex

248. Write short note on ***any three*** of the following:
i. LL(1) grammar
ii. Left Factoring
iii. Three Address Code
iv. Recursive Descent parsing
v. Ambiguous grammar

249. Write short note on ***any three*** of the following:
i. Predictive parsing
ii. Semantic Analyzer
iii. Syntax tree
iv. Quadruples
v. Peephole optimization

250. What do you mean by Peephole optimization? Explain with example the different ways of implementing Peephole optimization **5+10**