Question Bank

10CS63:Compiler Design

1.Determine whether the following regular expressions define the same language? (ab)* and a*b*

2.List the properties of an operator grammar
3. Is macro processing a phase in compilation? Justify your answer
4. Why is buffering used in lexical analysis? What are the commonly used buffering methods?
5. _____ verifies whether the input string can be generated from the grammar of the source language.
6. Write the subset construction algorithm
7. _____ parsers build parse trees starting from the root node and work down to
the leaves. These parsers satisfy _____ grammars
8. A _____is a program that collects statistics on the behavior of the object program
during execution

9. What are the types of attributes in an attributed grammar?
10. What optimization can you propose for the following code
a := b*c;
x := b*c +5;
11. Define text formatters.
12. Write three address code for the expression r := 9*2+7.
13. Write shorthand notation for rr*.
14. Define finite automata.
15. What is LL(1) grammar?
16. What is YACC?
17. What are the two types of attributes that are associated with a grammar symbol?
18. State different storage-allocation strategies.
19. Give example for algebraic transformation on basic blocks?
20. Name tow descriptors used in code generation algorithm?
21. _____ is a program that reads a program written in one language and translates it into an equivalent program in another language

22. What are the two classes of parsing methods?
23. Define Lexeme.
24. Write any two algebraic properties of regular expression.
25. _____ is a program consisting of a procedure for every non terminal.
26. What do you mean by ambiguous?

27. Define Annotated parse tree.
28. _____ keeps track of live procedure activations.
29. A loop that contains no other loops is called a _____.
30. Define optimizing compiler.
31. What is the purpose of semantic analysis in a compiler?
32. What are cousins of a compiler?
33. Draw NFA for regular expression (s/t).
34. What is meant by Lexeme?
35. List out bottom up parsing.
36. What is canonical derivation?
37. What is inherited attribute?
38. What is coercion?
39. What is constant folding?
40. What is a basic block?
41. List the functions of a preprocessor.
42. List the cousins of a compiler.
43. What language does the regular expression (0|1)*0(0|1)(0|1) generate?
44. Define e-closure.
45. What language does the grammar S →aSa | bSb | c generate?
46. Define operator grammar.
47. What is meant by quadruple?

48. What is called annotated parse tree?
49. Define flow graph.
50. Translate the arithmetic expression a * - (b + c) into a syntax tree.
51. Define Lexeme.
52. List two compiler construction tools.
53. What is meant by Kleene Closure?
54. Define optimizing compilers.
55. What language does the grammar S aSb | c generate?
56. Define ambiguity of a grammar.
57. What is synthesized attribute?
58. Give the postfix for a=b*-c+b*-c
59. Define constant folding.
60. What is data flow analysis?
61. Define Ambiguous grammar.
62. List out the cousins of compiler.
63. Define lexeme. Give example.
64. Compare DFA and NFA.
65. Define 'Handle Pruning' in bottom-up parsing.
66. What is meant by a Look ahead symbol?
67. What is three address code?
68. Define annotated parse tree.
69. Construct a DAG for the expression a=b*-c + b*-c
70. Define symbol table.

71. What is an interpreter?
72. List any two cousins of compiler.
73. Write a regular expression for an identifier.

74. List the various error recovery strategies for a lexical analysis.
75. Define a context free grammar.
76. What are the problems with top down parsing?
77. What are the various types of intermediate code representation?
78. What is back patching?
79. What is a flow graph?
80. What are the basic goals of code movement?
81. What is a byte code?
82. What is the use of an interpreter?
83. How is a character string recognized?
84. Define Lexeme.
85. List the common syntactic errors.
86. What do you mean by panic mode error recovery?
87. What is syntax directed definition?
88. What is the use of a dependency graph?
89. In a program, when the instruction HALT is given, where does the control return to?
90. Define semilattice.
91. What does the front end of a compiler do?
92. Write the components of context free grammar.
93. How is the C language identifier represented by a regular expression?
94. Write the purpose of minimizing the states of the DFA in a lexical analyzer.
95. List the properties of operator precedence grammar.
96. Mention the four values of LR parsing table.
97. Give an example of static checks.
98. Mention the strategies of storage allocation.
99. Write the various three address code form of intermediate code.

100. What is the various output form of the code generator?
101. What is meant by real time operating systems?
102. Compare loosely coupled and tightly coupled system.
103. What is meant by context switch?
104. What are the benefits of multithreaded programming?
105. What are the three requirements that a solution to the critical-section problem satisfy?
106. Define deadlock.
107. What is compaction?
108. How do you compute the effective access time for a demand-page system?
109. What is latency time?
110. What are streams?
111. What are the functions performed by analysis phase of a compiler?
112. Mention the components of context free grammar.

113. What are the two phases of lexical analyzer?
114. Write the regular expression for the language, the set of strings over {a,b,c} that contain exactly one b.
115. Define left factoring.
116. Mention the conflicts that occur in shift-reduce parser.
117. What are the functions used to create the nodes of the syntax trees for expressions with binary operators?
118. List out the methods available to represent the value of Boolean expression.
119. Name the techniques in loop optimization.
120. How are dummy blocks with no statements indicated in global data flow analysis?
121 . What are the two parts of compilation?
122. Define ambiguous grammar.
123. _____ represent strings of characters in the source program.

124. Write the regular expression for denoting the set containing the string 'a' and all
strings consisting of zero or more 'a''s followed by a b.
125. Why is error detection and recovery centered around syntax analysis phase?
126. What is parsing?
127. Mention the restrictions that are invoked by translation routines during parsing?
128. Why do we need backpatching?
129. List the benefits of using a machine independent form.
130. Define basic block.
131. Define Compiler.
132. List any two cousins of a Compiler.
133. "baan" is a subsequence of the string "banana". (True/ False)?
134. Write the regular expression for a sequence of binary numbers.
135. The rightmost derivation is known as _____.
136. When is a grammar said to be left recursive?
137. In a syntax directed definition if all attributes are synthesized, what is the name of
the syntax directed definition?
138. List the various static checks.
139. Give an example for a dangling reference.
140. Define induction variables.
141. What is parsing?
142. What do you mean by Cross-Compiler?
143. Define lexeme.
144. What is the role of lexical analysis phase?
145. A top-down parser generates
a. right-most derivation b. right-most derivation in reverse
c. left-most derivation d. left-most derivation in reverse

146 Context Free Grammar can be recognized by Push Down Automata.
(True/False)

147. Name some variety of intermediate forms.
148. Draw syntax tree for the expression a=b*-c+b*-c.
149. Mention some of the major optimization techniques.
150. Define basic block.
151. List the functions performed by analysis phase of a compiler.
152. Mention the components of context free grammar.
153. Write the regular expression for the language, the set of strings over {a, b, c} that
contain exactly one b.
154. Write the operations on NFA states.
155. Define left factoring.
156. List the various lexical errors.
157. Write the different storage allocation strategies.
158. List out the methods available to represent the value of Boolean expression.
159. List the two classes of transformations applied to basic blocks.
160. How to represent the dummy blocks with no statements indicated in global data
flow analysis?

PART-B Questions

1. Differentiate between lexeme, token and pattern
2. Draw the parse tree for the following natural language grammar:
Sentence à <Noun Phrase><Verb Phrase>
Noun Phrase à <Article><Noun>
Verb Phrase à <verb><Noun Phrase>
Article à an | a
Noun à <Adjective><Noun>
Adjective à huge
Noun à animal | elephant
Verb à is

3. Identify whether the following grammar is ambiguous:
G={(S),{a,b},{SàSaS,Sàb},S}
4. Is the following grammar left recursive? If so eliminate left recursion
S à Aa | b
A à Ac | Sd | e
5. What are the advantages of LALR parsing over SLR and CLR methods?
6. What do you meant by data-flow engine?
7. Draw the transition diagram for identifier.
8. Give example for reduction.
9. What is activation tree?
10. What is copy propagation? Give example.
11. Write a short note on functions of preprocessors.
12. What are the issues in lexical analysis?
13. Write a short note on left factoring.
14. Describe about type systems.
15. What are the different operations used to define semantic rules?

16. Draw a Parse tree for the assignment statement using appropriate grammar. position : = initial + rate* 60
17. Write a segment of code to move forward pointer of input buffer.
18. Consider the arithmetic expression grammar, show left most and right most derivation for id + id * id and draw their parse trees.
19. Write an algorithm for construction of dependency graph from a parse tree.
20. Give quadruples and triples for an assignment statement a := b*-c + b*-c.
21. What are the different data structures used for symbol table?
22. Compare syntax tree and parse tree.
23. Do left factoring in the following grammar.
A ->aBcC | aBb | aB | a

B ->ε

C ->ε

24. Define *l*-value and r-value.
25. What is meant by induction variable elimination?
26. Define Preprocessor. What are its functions?
27. What are the different strategies employed by parsers to recover from a syntactic error?
28. Remove left recursion in the following grammar.

S→AS | b

A→SA | a

29. Compare quadruples and triples.
30. What are registers and address descriptors?
31. Write a short note on input buffering techniques.
32. What are the properties of Regular Expression?
33. Discuss about the 'panic mode' error recovery strategies of the parser.
34. Mention the different storage allocation strategies.
35. Write shortly about DAG (Directed Acyclic Graph) representation.
36. What are the phases that constitute the front end of a compiler?
37. Write the algorithm for FOLLOW.
38. What is shift – reduce conflict?
39. Give the syntax-directed definition for if-else statement.
40. What is a basic block? Give an example.
41 List the steps for a language processing system.
42. Differentiate lexical analysis and parsing.
43. Draw the parse tree for the arithmetic expression id + (id * id).
44. How is the syntax tree constructed?
45. What does data flow analysis frame work consist of?
46. What are the functions of structure editor?
47. Why are regular expressions used to define lexical syntax of a language?

48. What are the different types of errors a program can contain? List out the error handling strategies.

49. Write an algorithm to partition a sequence of three-address statements into basic
blocks.
50. Write about type systems and checking of types.
51. List any three services provided by an operating system. Explain how each provides convenience to the users.
52. Compare short-term, medium-term, and long-term schedulers.
53. Summarize the Safety Algorithm.
54. What are the various scheduling criteria for CPU scheduling?
55. What is low-level formatting?
56. What are the functions performed when the preprocessors produce input to compilers?
57. Define NFA and DFA.
58. Show the techniques available to construct an LR parsing table for a grammar and features.
59. Draw the syntax tree and postfix notation for the expression a: =b*-c
60. Draw the DAG for the expression a: = b *-c + b*-c.
61. What is the function of Query interpreters?
62. List the issues involved in lexical analysis.
63. Write an algorithm to eliminate left recursion from a grammar.
64. What are the three storage allocation strategies?
65. What are the difficulties in instruction selection?
66. Explain about Assembler with an example.
62. E à E + E | E * E |(E) | id. Write the leftmost derivation and rightmost derivation for the sentence (id*id)+id.
63. Write the algorithm to find the FIRST of nono-terminals.
64. What is an Activation record?
65. Draw the DAG for the statement a=(a*b+c)-(a*b+c).

66. Write a note on syntax directed translation.
62. Write state program for the token, "id-identifier".
63. Define left recursion. Eliminate left recursion from the grammar. S ® Aa/b, A ® Ac/Sd/e?
64. What are the three kinds of intermediate representations? Explain them.
65. What is the step takes place in peephole optimization? What are the characteristics of peephole optimization?
66. Write the benefits of intermediate code generation.
67. Differentiate NFA and DFA.
68. Write about the role of the parser.
69. Draw the syntax tree and postfix notation for the expression a: =b*-c.
70. Write about type systems and checking of types.

PART –C Questions

1. Explain in detail the process of compilation. Illustrate the output of each phase of compilation for the input "a = (b+c)* (b+c) * 2"
2. a. Define the flowing terms Compiler, Interpreter, Translator and differentiate between them.

b. Why is it necessary to study the theory behind the design of compiler.
3. Construct the NFA, DFA and minimized DFA for the regular expression have single line comments with characters from the alphabet {a,b}.
4. a. What is the role of the lexical analyzer? Explain
b. Identify the token and, lexemes in the following function:

```
functiongcd (m, n: integer): integer;
begin
if n = 0 then gcd := m
elsegcd := gcd (n, m mod n)
end; ( * of gcd*)
```

5. a. Write the algorithm to carry our operator precedence parsing action and Explain

b. Construct the operator precedence parse table for the following grammar and show its shift-reduce actions for the input string " abab".

S →aSbS | bSaS | ε

6. Consider the following grammar

S → AS | b

A → SA | a

Construct the SLR parse table for the grammar. Show the actions of the parser for the input string "abab".
7. a. Compare the different implementations of three address codes with examples
b. Are the attributes in the following CFG synthesized or inherited? Give reasons:

Var→IntConstant

{$0.*val* = $1.*lexval*;}

Expr→Var

{$0.*val* = $1.*val*;}

Expr→Expr B-op Expr

{$0.*val* = $2.*val*($1.*val*,$3.*val*);}

B-op → +

{$0.*val* = *PLUS*;}

B-op → *

{$0.*val* = *TIMES* ;}

8. Describe the syntax directed translation procedure for assignment statements with
integers and mixed types and explain.
9. Construct the TAC and draw the flow graph for the following procedure.

```
void quicksort(int m, int n)
{
int i, j;
int v, x;
```

```
if (n <= m) return;
i = m-1; j = n; v = a[n];
while(1) {
do i = i+ 1; while (a[i] < v);
do j = j+ 1; while (a[j] > v);
if( i >= j) break;
x = a[i]; a[i] = a[j]; a]j] = x;
}
x = a[i]; a[i] = a[n]; a]n] = x;

quicksort(m, j); quicksort(i+1, n);
}
```

Identify and eliminate induction variables in the same.

10. Discuss the issues in code generation with examples.
11. Explain the various phases of compiler in detail with suitable example.
12. a. Explain program development environment. b. What are the two parts of compilation?
13. Obtain the minimized state DFA for the regular expression (a/b)*a using subset construction method.
14. Write the algorithm to convert from NFA to DFA.
15. Construct the predictive parser for the following grammar

E ->TE'
E' ->+TE' | e
T ->FT'
T' ->*FT' | e
F ->(E) | id

16. a. Describe the different error recovery techniques.
b. What is left recursive grammar? How will you eliminate? Give example.
17. Explain the Syntax directed definition for construction of syntax tree with example.
18. Explain different type expressions with example.
19. Explain the principal sources of optimization with example.
20. Explain the issues of code generator.
21. Explain different phases of compiler in detail.
22. Write short notes on the following:
a. Compiler – construction tools
b. Precedence of operators
23. Explain how to incorporate a symbol table.
24. Describe about Nondeterministic and Deterministic finite automata.
25. Explain non recursive predictive parsing in detail.