SSCD-Super Important questions-18CS61

By the TIE review team DSATM, BNMIT and SVIT

Module-1

1. With suitable examples, explain the SIC/XE Machine Architecture and its I/O operation
2. What is program relocation? Explain the problem associated with it and the solutions.
3. Write the algorithm for a pass-1 of two pass assembler and pass 2 of two pass assembler - (Practice anyone)
4. Explain the following: Literals, Symbol defining statements, Expressions
5. Apply the concept of the control section to a suitable code snippet and explain how the control section helps us achieve independent assembling of the source program
6. Explain the following -15M
   (a) Record operations-3M (b) Format of Header,text,end record-3M
   (c) Program blocks -3M (d) defn of SYMTAB,LOCCTR,OPTAB-3M
   (e) Diff between one pass and multipass assembler-3M

Module-2

7. Illustrate the structure of a compiler with a neat diagram, Show the output of each phase of the compiler for the assignment statement:

   Profit= Sellingprice-costprice*90
8. Explain the various applications of compiler technology
9. Apply input buffering strategy to a suitable example and explain the same, write an algorithm for lookahead code with sentinels
10. Explain the following -3M each

(i)Interpreter v/s assembler (ii)Language processing System (iii)Phase v/s pass
(iv)Lexical analysis (role)

11. Design a lexical analyzer to identify the keywords begin, else, and end. Sketch the program segment to implement it showing the first two states and one final state

12. What is the meaning of patterns, lexemes and tokens? Identify lexemes and tokens in the following statement: a= b*d;

Module-3

13. Build an algorithm to eliminate left recursion. Eliminate left recursion from the following grammar

(a)$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

(b) $S \rightarrow Aa \mid b$

$A \rightarrow Ac \mid Sd \mid \varepsilon$

14. What are the different sentential forms? What is the left sentential form? What is the right sentential form?

15. Explain the process of identifying ambiguous grammar, check whether the grammar below is ambiguous and if yes eliminate the ambiguity.

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

$C \rightarrow b$

16. Given the following grammar:

$Z \rightarrow d \mid XYZ$

$Y \rightarrow \varepsilon \mid c$

$X \rightarrow Y \mid a$

a. Compute FIRST and FOLLOW sets for each non-terminal

b. Without constructing the parsing table, check whether the grammar is LL(1).

c. By constructing the parsing table, check whether the grammar is LL(1).

17. Consider the following grammar:

$E \rightarrow TE^1$
$E^1 \rightarrow +TE^1 \mid \in$
$T \rightarrow FT^1$
$T^1 \rightarrow *FT^1 \mid \in$
$F \rightarrow (E) \mid id$

a) Identify FIRST and FOLLOW b) Construct the predictive parsing table
c) Show the sequence of moves made by the parser for the string id+id*id
d) Add the synchronizing tokens for the above parsing table and show the sequence of moves made by the parser for the string ") id * + id"

18. Consider S -> SS+ |SS* |a, and input string aa+a*, and do the following
a)Identify LMD and RMD b)Check whether the grammar is ambiguous (iii)Do Left factoring and solve for left recursion if any

19. Given the following grammar:
S → a | (L)
L → L, S | S
a. Is the grammar suitable for predictive parser?
b. Do the necessary changes to make it suitable for LL(1) parser
c. Compute FIRST and FOLLOW sets for each non-terminal
d. Obtain the parsing table and check whether the resulting grammar is LL(1) or not.
e. Show the moves made by the predictive parser on the input "(a, (a , a))"

Module-4

20. (a) Write a LEX program to count the number of scanf and printf statements and replace them with readf and writef respectively.
(b)Write a LEX program to identify vowels and constants

21. What is ambiguous grammar? Explain the problem of arithmetic ambiguity by considering the expression **2+3*4**, Make use of precedence and associativity rules to resolve the ambiguity

22. Explain/Write the following wrt YAAC-12M
(i)Types of conflicts with ex (ii)Errors handled by YACC (iii) A YACC program to check whether the given arithmetic expression is valid or not. (6M)

22. Explain the metacharacters used in a regular expression with an example.

Module-5

–23.Obtain SDD for the following grammar using a top-down approach:
S → En
E → E + T | T
T → T * F | F
F → (E) | digit
and Obtain annotated parse tree for the expression (3 + 4) * (5 + 6)n

24. Define SDD, Synthesized and inherited attributes with examples, Analyse and bring out the distinct differences between S attributes and L attributes

25. Write Semantic rules(refer tie notes) and a dependency graph for the Input string, based on the SDD given below: float id1,id2,id3
D → T L
T → int
T → float
L → L1 , id
L → id

26. (a) Analyse the expression given and write the three address form and DAG for a + a * (b – c) + (b - c) * d
(b)explain the three address form and apply this to form quadruple representation and triples and indirect triples for a= c *-d+ c *-d
(c)Write the SDT for converting an infix to prefix expression. Show the actions for translating the expression 2*3+4 into its equivalent prefix expression

27. Explain the following with an example
(a)Semantic rule (b)Synthesized attribute (iii)inherited attribute (iv)Annotated parse tree

How we frame these questions:
TIE review team frames these questions by consolidating comprehensively from the following sources
1. Exercise problems of textbooks/ references
2. Previous year question VTU exam Question paper and MQP
3. Questions by Various HODs and professors in contact with TIE

Questions framed shall follow all Bloom's learning levels with appropriate action verbs ensuring coverage of all COs

Notes available for all modules:
https://takeiteasyengineers.com/category/cse-ise/6th-sem-cse-ise/sscd/