Pune Vidyarthi Griha's

COLLEGE OF ENGINEERING, NASHIK – 4

COMPUTER ENGINEERING DEPARTMENT

Subject : COMPILER **ASSIGNMENT NO – 02** **Unit : II**

1. **Write the rules to calculate FIRST( ) and FOLLOW( ) function?**
2. **Explain Error Recovery strategy in Parser?**
3. **Explain the role of Parser. State interaction between lexical analyser and parser?**
4. **Compare Recursive decent and Predictive parser.(4 points atleast)**
5. **Explain the operator precedenence parser.**
6. **Differentiate between SLR, LR(K) and LALR Parser.(4 points atleast)**
7. **Differentiat between topdown and bottom up parsing.(4 points atleast)**
8. **Explain automatic contruction of parser using YACC?**
9. **Explain what is elimination of Left recursion and Left factoring in Predictive parsing?**
10. **Find FIRST and FOLLOW sets for given grammar.**

    S -> PQR

    P -> a | Rb | €

    Q -> c | dp | €

    R -> e | f

11. Compute FIRST and FOLLOW for the following grammar

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

12.

Consider the grammer E ->E - E,E-> E*E,E -> id
Write shift reduce parsing steps of the input string "id1-id2*id"

13.

Test whether following grammar is LL (1)
S → i E t S S' | a
S' → eS | empty,
E → b

14. Generate SLR Parsing table for the given grammar

S → OSO | 1S1 | 10

15. Generate SLR Parsing table for the given grammar and parse the string id1 * id2 + id3

E → E + T | T

T → T * F | F

F → (E) | id

16. Show that following grammar is LL(1)

S → AaAb | BbBa

A → €

B → €

17. Construct LL(1) parsing table foe following grammar

S → aB | aC | Sd | Se

B → bBc | f

C → g

18.For following grammar

S → AaBb
A → €
B → €
I) Computer First and Follow Sets.
II) Construct LL(1) Parser
III) Parse string "ab" with the above parser.

19.Generate the predictive parsing table for given grammar and parse the string acbbgf.

S → aBDh

B → Bb | c

D → EF

E → g | €

F → f | €

20. What is LL(1) grammar?

Compute the FIRST () & FOLLOW () of following grammar :

S → aAbCD | €

A → ASD | €

B → SaC | hC | €

C → Sf | Cg

D → aBD | €

21.Design SLR parsing table for given grammar:

S → AS|b

A → SA|a

22.Check whether the following grammer is LL(1)

S → aABb
A → c | €
B → d | €

23. Construct LR(1) parsing table for following grammar :

S → Aa | bAc | Bc | bBa

A → d

B → d

24. Construct SLR Parser for following grammar

S → aSSb

S → aSSS

S → c

Show moves of above parser on one valid input string and one invalid input string.

25. Justify the role of stack in the design of Bottom-up parser. Verify whether following **grammar is SLR**

S → L = R

S → R

L → *R

L → id

R → L

26. Construct SLR parser for the following grammar

S → aSA | €

A → bB | cc

B → bd | €

Parse the sentence "aabbdcc" with the above SLR Parser.

NOTE :- Above Bold text question are important for In-sem examination

********** Best of Luck **********

Prepared BY : Prof. Gharu Anand