Total number of printed pages–7

53 (IT 603) CPDG

2018

COMPILER DESIGN

Paper : IT 603

Full Marks : 100

Time : Three hours

The figures in the margin indicate full marks for the questions.

*Answer **any five** out of **eight** questions.*

1. *(a)* Explain with a neat diagram, the various phases of a compiler and explain how different phases of compilation will operate on the following statement

position = initial + rate*60

assuming data type of rate is float. 6+4

(b) What is the relationship with lexical analyzer, regular expressions and transition diagram ? Give an example. 6

(c) What is syntax directed translation and why they are important ? 2+2

Contd.

2. *(a)* Define the following term Lexeme Lexical analyzed and token. 6

(b) What do you mean by boot strapping process? What is the advantage of using this process? 3+3

(c) Remove left Recursion from the following grammar 8

$exp \rightarrow exp \;\; addop \; term \mid term$

$addop \rightarrow + \mid -$

$term \rightarrow term \;\; mulop \; factor \mid factor$

$mulop \rightarrow *$

$factor \rightarrow (exp) \mid number$

where +, − , (,), *, number are terminals.

3. *(a)* Compute FIRST and FOLLOW from the grammar below 6

$S \rightarrow SAB \mid SBC \mid \varepsilon$

$A \rightarrow aAa \mid \varepsilon$

$B \rightarrow bB \mid \varepsilon$

$C \rightarrow cC \mid \varepsilon$

(b) How do you check whether a grammar is $LL(\perp)$ or not? Check whether the grammar given below is $LL(\perp)$ or not. 3+3

$S \rightarrow aABb$

$A \rightarrow c \mid \varepsilon$

$B \rightarrow d \mid \varepsilon$

(c) Construct the transition diagram for following regular expressions: 2+3+3

(i) $(a \mid b)^*$

(ii) $((a \mid b)c^*)^*$

(iii) $(a \mid b)^* abbb$

4. *(a)* Define the following: 2×3

(i) Parse tree

(ii) Left most derivation

(iii) Right most derivation.

(b) Construct CLR parse table for the following augmented grammar: 8

$S' \rightarrow S$

$S \rightarrow Cc$

$C \rightarrow Cc \mid d$

(c) Make left and right most derivation using top down and bottom up strategy to derive a statement W. Where $W = id + (id + id) * id$ using the following grammar : 6

$$E \rightarrow E + E$$

$$E \rightarrow E * E$$

$$E \rightarrow (E)$$

$$E \rightarrow id$$

5. *(a)* Create a DAG for the expression below : 6

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

(b) Write the 3-address code, quadruple, triple and indirect triple for the expression

$$(x + y) * (y + z) + (x + y + z)$$ 9

(c) Why is CFG important in the syntax analysis phase of compiler? 5

6. *(a)* Consider the following grammar given below and construct the LALR parsing table. Consider the augmented grammar $G^{\perp}$ 10

$S^{\perp} \to S$

$S \to aAd \mid bBd \mid aBc \mid bAc$

$A \to C$

$B \to cb$

(b) Explain syntax directed translation scheme with example. 5

(c) Differentiate between L-attributed and S-attributed grammar? 5

7. *(a)* Consider the context free grammar below: 8

$S \to EN$

$E \to E + T \mid E - T \mid T$

$T \to T * F \mid T / F \mid F$

$F \to (E) \mid digit$

$N \to j$

(i) obtain SDD for the above grammar

(ii) construct, the parse tree, syntax tree and annotated parse tree for the input string $10 + 5 * 3$.

(b) What is Handle pruning? In which parser it is used? 4

(c) What do you mean by left factoring? What is its use in parsing? Do the left factoring the following grammar: 2+2+4

$E \rightarrow 5 + T \mid 3 - T$

$T \rightarrow V \mid V * V \mid V + V$

$V \rightarrow a \mid b$

8. *(a)* Briefly explain the problems associated with top down parser? 5

(b) Consider the following grammar

$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow (E)$

$E \rightarrow id$

Using the above grammar for input string $id1 + id2 * id3$. Show the stack implementation for shift reduce parsing. 6

(c) Write short notes on : ***(any three)*** 3×3

(i) LEX

(ii) Recursive descent parsing

(iii) Global register allocation

(iv) Panic mode error recovery

(v) Symbol table.