53 (CS 601) CPDG

2017

COMPILER DESIGN

Paper : CS 601

Full Marks : 100

Time : Three hours

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

*Answer questions **1** and **2** and **any two** from the rest.*

1. Consider the following grammar rules

   $A \rightarrow AaBX \mid B$

   $B \rightarrow BbY \mid Y$

   $X \rightarrow x$

   $Y \rightarrow (A) \mid y$

   (a) remove left recursion for the above grammar.

   (b) compute FIRST and FOLLOW on the rules you got in Question 1(a). (i.e after removal of left recursion)

(c) check whether the grammar is LL (1) or not.

(d) construct a predictive parsing table.

(e) check whether the string "yayx" will be accepted or not. 5+10+5+5+5

2. Consider the following augmented grammar

$S' \rightarrow S$

$S \rightarrow xYZ$

$Y \rightarrow yZ \mid p$

$Z \rightarrow q$

(a) construct the set of LR (1) items

(b) design a canonical-LR (1) parsing table

(c) construct LALR parsing table. 15+10+5

3. (a) Consider the following instruction

$$x = y + z/30 - p$$

clearly mention the different phases (with the help of diagrams) of Compiler.

(b) remove left factoring

$S \rightarrow ABcD \mid ABXY,\ D \rightarrow d,\ X \rightarrow x,\ Y \rightarrow y$

$A \rightarrow a,\ B \rightarrow b$

discuss the problems of top-down parsing if left factoring present in a grammar. 10+10

4. Consider the following regular expression

$(p \mid q)^*pq(r)^*$

(a) Design a NFA for the expression.

(b) Construct equivalent DFA.

(c) Minimize the number of states. 5+10+5

5. Consider the following expression

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

(a) construct DAG for the expression

(b) write equivalent three address code

(c) represent your three address code into quadruples and triples

(d) minimize your three address code. 5+5+5+5
