53 (IT 603) CPDI

2014

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** questions out of **seven**.*

1. *(a)* Explain with neat diagram, the various phases of compiler. Mention the input and output for each phase. 6+4

   (b) What is the role of lexical analyzer ? What are lexemes, tokens and patterns. 4+6

2. *(a)* What is Syntax Directed Translation (SDD) and why it is important ? 3+2

   (b) Define the following : 3×3

   (i) Parse tree

   (ii) Left-most derivation

   (iii) Right-most derivation

(c) Give Left-most and Right-most derivation to derive a statement W. 4+2

$W = id + (id) + id * id$ by using the following grammar.

$\dot{E} \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow (E)$

$E \rightarrow id$

Check whether the above given grammar is ambiguous for the statement W, or not.

3. (a) Draw the DAG for the following expression 4+6

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

also translate the above expression into 3-address code, quadruples and triples.

(b) Construct the LR(0) Parsing table for the following grammar 6+1

$S \rightarrow L = R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow id$

$R \rightarrow L$

Is the above grammar LR (0) ?

(c) Write down the differences between top-down and bottom-up parsing method. 3

4. (a) Compute the FIRST ( ) and FOLLOW ( ) symbols for the following grammar. 6

$$E \rightarrow TE'$$
$$E' \rightarrow +TE \mid \varepsilon$$
$$T \rightarrow FT'$$
$$T' \rightarrow *FT' \mid \varepsilon$$
$$F \rightarrow (E) \mid id$$

(b) Define LL (1) grammar. Under what conditions a grammar is called LL (1)? Check whether the above grammar (question no. 4 (*a*)) is LL (1) or not. 2+3+3

(c) Differentiate between L-attributed and S-attributed grammar. 6

5. (a) What do you mean by left factoring ? What is the use of it in parsing ? Do the left factoring of the following grammar 2+2+4

$$E \rightarrow 5+T \mid 3-T$$
$$T \rightarrow V \mid V*V \mid V+V$$
$$V \rightarrow a \mid b$$

(b) Explain how stack implementation of shift-reduce (SR) parsing is done considering the following grammar. 8

$E \rightarrow E + E$
$E \rightarrow E * E$
$E \rightarrow (E)$
$E \rightarrow id$

the input string is $id + (id) + id * id$.

(c) What are the advantages and disadvantage of LALR Parsing technique ? 4

6. (a) Explain left recursion. Describe the algorithm used for eliminating the left recursion. 2+4

(b) Eliminate left recursion from the following grammar 6

$S \rightarrow aB \mid aC \mid Sd \mid Se$
$B \rightarrow bBc$
$C \rightarrow g$

(c) Consider the following grammar and construct SLR parsing table 8

$E \rightarrow T + E \mid T$
$T \rightarrow int * T \mid int \mid (E)$

Also show the actions of the parser for the input string "$int * (int) + (int * int)$".

7. *(a)* What is handle Pruning ? Explain with the help of the grammar $S \rightarrow SS+ \mid S*S \mid a$ and input string "$aa+*aa+$". 2+4

(b) What do you mean by boot strapping process ? What is the advantage of using this process ? 4

(c) Consider the following grammar given below and construct the LALR Parsing table. Consider the augmented grammar G'. 10

$$S' \rightarrow S$$
$$S \rightarrow Aa$$
$$S \rightarrow dAb$$
$$S \rightarrow dCa$$
$$S \rightarrow cb$$
$$A \rightarrow c$$
