44 (5) CODG 5.2

2009

COMPILER DESIGN

Paper : 5·2

Full Marks : 80

Time : Three hours

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

1. (a) State the different phases of the front end of a compiler and give a brief description of the work of each phase. 5

(b) Define linker and loader. 3

2. From the regular expression $a^*b^+(aba)$ generate a NFA first and then convert it into a DFA. 8

Or

Write an algorithm to minimize the number of states of a DFA. 8

Contd.

define *NFA* and *DFA*. Explain how an *NFA* can be converted to a *DFA*. 4+16=20

Or

Write lexical analyser for identifier, the operators +, *, >, >=, <, <=, ==, the keywords int, for, while, return and string constant as in C-language. 20

3. Considering the following grammar :

$E \rightarrow TE'$

$E' \rightarrow +TE'/\in$

$T \rightarrow FT'$

$T' \rightarrow *FT'/\in$

$F \rightarrow (E)\ id$

(a) Give FIRST and FOLLOW set. 3+3=6

(b) Construct predictive parsing table. 8

(c) Show the stack moves made by predictive parser on input $id + id * id$. 6

4. Answer ***either*** [(a), (b) and (c)] ***or*** [(d) and (e)]

(a) What is basic block ? Give algorithm for basic block. 2+8=10

(b) What is flow graph ? Explain. 5

(c) What is LALR parser ? Explain. 5

(d) What is parse tree ? Explain with an example. 5

(e) Considering the grammar

$S' \rightarrow S$

$S \rightarrow CC$

$C \rightarrow cC/d$

construct canonical *LR* parsing table. 15

5. Answer ***either*** (a) ***or*** (b) and (c).

(a) Show how flow of control statements given by the following grammar (with usual meaning) can be translated :

$S \rightarrow$ if E then S

| if E then S else S

| while E do S

| begin L end

| A

$L \rightarrow L$; S

| S

A is an assignment statement and E is a Boolean expression (make necessary assumptions). 20

Or

(b) What is *DAG* ? Draw a *DAG* for the three address code 2+8=10

i) $t_1 := 4 * i$

ii) $t_2 := a[t_1]$

iii) $t_3 := 4 * i$

iv) $t_4 := b[t_3]$

v) $t_5 := t_2 * t_4$

vi) $t_6 := prod + t_5$

vii) $prod := t_6$

viii) $t_7 := i+1$

ix) $i := t_7$

x) if $i <= 20$ goto *(i*

(c) Explain the code generation algorithm. 10

6. (a) What is left factoring ? Explain with an example. 5

(b) What do you mean by recursive descent parsing ? Explain. 5

(c) What is three-address statement ? In how many ways these statements can be represented ? Explain. 2+8=10

7. (a) What is type checking ? Explain the implementation of syntax-directed translation with an example. 2+8=10

(b) Distinguish between top-down and bottom-up parsing. 5

(c) What is regular expression ? Explain. 5

8. Write short notes on *any four* of the following : 5×4=20

(a) Linker

(b) Elimination of dead code

(c) Storage management

(d) Contents of symbol table

(e) Syntax tree

(f) Operator precedence parsing

(g) Activation Trees.