2008

COMPILER CONSTRUCTION

PAPER—2403

Full Marks: 70

Time: 3 hours

Answer any five questions

The figures in the right-hand margin indicate marks

Candidates are required to give their answers in their own words as far as practicable

Illustrate the answers wherever necessary

1. (a) Consider the grammar

   \[ S \rightarrow (L) / a \]

   \[ L \rightarrow L, S / S \]

   Find the parse trees for the following sentences:

   (i) \((a, (a, a))\)

   (ii) \((a, ((a, a), (a, a)))\)
(2)

(b) (i) What is left recursion?

(ii) Eliminate the left recursion for the above grammar.

(iii) What is left factoring?  

\[(3 + 5) + (2 + 2 + 2)\]

2.  (a) What is ambiguous grammar? Test the following grammar w.r.t ambiguity:

\[
S \rightarrow aB/ab
\]

\[
A \rightarrow aAB/a
\]

\[
B \rightarrow ABb/b.
\]

(b) Briefly describe the different phases of compiler. What is a cross compiler.

\[(2 + 3) + (7 + 2)\]

3. Construct a predictive parsing table for the following grammar:

\[
S \rightarrow ACB/CbB/Ba
\]

\[
A \rightarrow da/BC
\]

\[
B \rightarrow g/\epsilon
\]

\[
C \rightarrow h/\epsilon.
\]
4. (a) Define LL(1) grammar. Is the following grammar LL(1)?

\[ S \rightarrow a \ A B b \ C D / \epsilon \]
\[ A \rightarrow A S d / \epsilon \]
\[ B \rightarrow S A c / h C / \epsilon \]
\[ C \rightarrow S f / C g \]
\[ D \rightarrow a B D / \epsilon \]

(b) Critically comment on the design and usefulness of 'Top-down Parser'.

(3 + 6) + 5

5. Construct SLR(1) parsing table for the following grammar:

Stat \rightarrow \text{if cond then Stat else Stat}

/ \text{if cond then Stat}

/ all other production for statement

6. Is the following grammar LR(1)? If yes, construct a Canonical parsing table:

\[ S \rightarrow a A \ d / b B d / a B e / b A e \]
\[ A \rightarrow C \]
\[ B \rightarrow C \]
7. Write short notes (any two):

(i) Basic block
(ii) Code optimization
(iii) Error handler
(iv) Intermediate code.