2008

AUTOMATA AND COMPILER

(*Theory of Computation and Compiler*)

PAPER— CS/MSC/1202

Full Marks : 40

Time : 2 hours

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

MODULE—M1

[*Marks* : 20]

Answer any *four* questions

1. Show that $L = \{a^n : n \text{ is a prime number}\}$ is not regular. 5

2. Construct a deterministic finite automata equivalent to the grammar

$$S \rightarrow aS \mid bS \mid a^{A}, A \rightarrow bB, B \rightarrow aC, C \rightarrow \wedge.$$ 5

3. Convert the following grammar into CNF 5

$$E \rightarrow E + E, E \rightarrow E * E, E \rightarrow (E), E \rightarrow 2$$

4. Convert the following grammar into GNF 5

$$S \rightarrow AB, A \rightarrow BS \mid b, B \rightarrow SA \mid a.$$

5. Construct a PDA A accepting the set of all strings over (a, b) with equal number of a's and b's. 5

6. Construct a DFA for equivalent regular expression $r = (01 + 2^*)^*.1$. 5

7. Construct a regular grammar G generating the regular set $r = 01^* (0+1)^*$. 5

MODULE—M2

[*Marks* : 20]

Answer any *four* questions

1. Find the FIRST & FOLLOW for the following grammar: 5

$S \rightarrow iEtSS' | a$
$S' \rightarrow eS | \wedge$
$E \rightarrow b.$

2. Consider the following grammar and test whether the grammar is LL(1) or not: 5

$S \rightarrow 1AB | \wedge$
$A \rightarrow 1AC | OC$
$B \rightarrow OS$
$C \rightarrow 1$

3. Show that no ambiguous or left-recursive grammar can be LL(1). 5

4. Generate the three address code for the following C progam: 5

```
if  a < b then
while  c > d do
      x = x + y
else
  do
        p = p + q
while I < = f
```

5. What is DAG? Show the DAG representation of Basic block with example. 1 + 4

6. What is cross compiler? Explain the bootstrapping technique. 1 + 4

7. Write short notes (any *two*): $2\frac{1}{2} \times 2$

   (*i*) Code optimization

   (*ii*) Dependency graph

   (*iii*) Type checking.