Roll No ..

CS-505

B.E. V Semester

Examination, June 2016

Theory of Computation

Time : Three Hours

Maximum Marks : 70

Note: i) Answer five questions. In each question part A, B, C is compulsory and D part has internal choice.

ii) All parts of each question are to be attempted at one place.

iii) All questions carry equal marks, out of which part A and B (Max. 50 words) carry 2 marks, part C (Max. 100 words) carry 3 marks, part D (Max. 400 words) carry 7 marks.

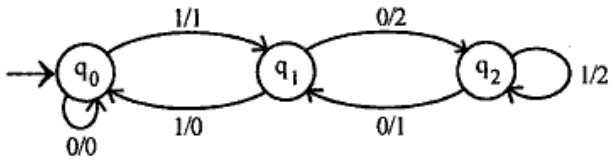
iv) Except numericals, Derivation, Design and Drawing etc.

Unit - I

a) Design DFA that accepts all strings with at most 3 a's.

b) Design a NFA for $\left\{cbab^n / n >= 0\right\}$.

c) Construct Moore machine for the following Mealy machine.



d) Write and explain Myhill-Nerode theorem.

OR

Construct NFA for the following grammar

$S \rightarrow Ab / ab, A \rightarrow Ab / Bb, B \rightarrow Ba / a$

Unit - II

2. a) Give CFG for R.E (011+1)*(01)*.
   b) Explain GNF conversion steps.
   c) Explain ambiguous grammar problem.
   d) Convert following CFG to CNF

$S \rightarrow ASB/E$

$A \rightarrow aAS/a$

$B \rightarrow SbS/A/bb$

OR

Convert the following grammar G into GNF

$S \rightarrow XA/BB$

$B \rightarrow b/SB$

$X \rightarrow b$

$A \rightarrow a$

Unit - III

3. a) Explain PDA.
   b) Explain how many way's PDA can accept (final out null store).
   c) Explain pumping lemma for CFL.

Contd...

   d) Design push down automata which accepts $L = \{0^n 1^{2n} / n \geq 1\}$

OR

Design a push down automata which accepts set of balanced parentheses. $(\{\{\{(\ )\}\}\})$

Unit - IV

4. a) Explain ID of a turing machine.
   b) Explain Multi Tape and Universal Turning machine.
   c) Explain church hypothesis.
   d) Design turing machine to add two numbers a and b.

OR

Design turning machine for accepting strings of the language defined as $\{\omega\omega\gamma / \omega \in (0+1)*\}$.

Unit - V

5. a) Explain P and NP problems.
   b) Difference between NP complete w NP hard problem.
   c) Explain process of Reducibility.
   d) Describe Hamiltonian path problem.

OR

Describe vertex cover problem.

******