

T.E. (Computer) (Semester – I) Examination, 2010
THEORY OF COMPUTATION
(2008 Course) (New)

Time: 3 Hours

Max. Marks: 100

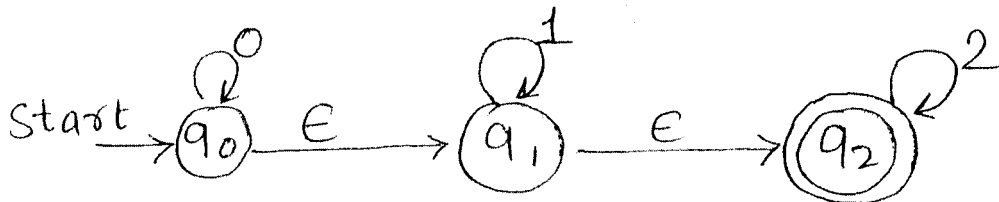
- Instructions :**
- 1) Answer **three** questions from **each** Section.
 - 2) Answers to the **two** Sections should be written in **separate** answer books.
 - 3) **Neat** diagrams must be drawn **whenever** necessary.
 - 4) Figures to the **right** indicate **full** marks.
 - 5) Assume suitable data, **if** necessary.

SECTION – I

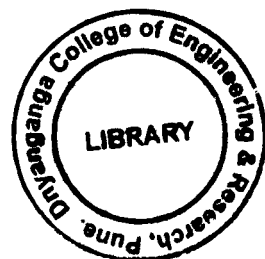
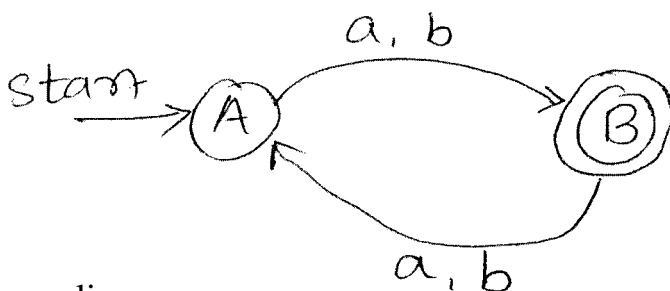
1. a) Define the following terms with example. 6
 - 1) Symbol
 - 2) Alphabet
 - 3) DFA
- b) Design a Mealy machine that accepts strings ending in '00' and '11'. Convert the Mealy machine to the equivalent Moore machine. 10

OR

2. a) Define the following terms with example. 6
 - 1) Basic machine
 - 2) Moore Machine
 - 3) Mealy machine
- b) Consider the following NFA with E-transitions. Convert this NFA to 10
 - 1) NFA without ϵ -moves
 - 2) DFA



3. a) Consider the following transition diagram. Convert it to the equivalent regular expression. 6



- b) Find a regular expression corresponding to each of the following subsets of $\{0, 1\}^*$. 6
- 1) The language of all strings containing exactly two 0's.
 - 2) The language of all strings containing atleast two 0's.
 - 3) The language of all strings not containing the substring 00.
- c) Prove the formula. 4
- 1) $\phi^* = \epsilon$
 - 2) $(r^* s^*) = (r + s)^*$

OR

4. a) For the following regular expression, draw an FA recognizing the corresponding language. 6

$$r = (1+10)^* 0$$

- b) Consider the two regular expressions

$$r = 0^* + 1^* \quad S = 01^* + 10^* + 1^*0 + (0^* 1)^*$$

- 1) Find a string corresponding to s but not to r.
 - 2) Find a string corresponding to both r and s.
 - 3) Find a string in $\{0, 1\}^*$ corresponding to neither r nor s. 6
- c) Explain the use of regular expressions in unix with any one example. 4
5. a) Simplify and convert the following CFG to Chomsky Normal Form. 12
- $$S \rightarrow AACD$$
- $$A \rightarrow aAb/\epsilon$$
- $$C \rightarrow aC/a$$
- $$D \rightarrow aDa/bDb/\epsilon$$

- b) Consider the CFG with productions

$$S \rightarrow S + S/S - S/S^*S/S/S / (S)/a$$

Is the grammar ambiguous ? If 'Yes' find the equivalent unambiguous grammar. 6



6. a) Describe the language generated by each of these grammars. Justify your answer with an example. 12
- 1) $S \rightarrow a S a \mid b S b \mid \epsilon$
 - 2) $S \rightarrow a S a \mid b S b \mid a \mid b$
 - 3) $S \rightarrow a S b \mid b S a \mid \epsilon$
 - 4) $S \rightarrow SS \mid bS \mid a$
- b) Convert the following grammar to Griebach Normal form. 6
- $S \rightarrow ABA \mid AB \mid BA \mid AA \mid A \mid B$
 $A \rightarrow aA \mid a$
 $B \rightarrow bB \mid b$

SECTION – II

7. a) Construct pushdown automata for $L = \{ W C W^R / W \in (a + b)^* \}$ and W^R is reverse string of W . 6
- b) Construct pushdown automata for the given CFG with productions 6
- $S \rightarrow a AA$
 $A \rightarrow bS \mid aS$
 $S \rightarrow a$
- c) Explain the equivalence of PDA with acceptance by final state and empty stack. 6

OR

8. a) Consider the PDA with following moves. Construct a CFG equivalent to PDA. 8
- $M = (\{q_0, q_1\}, \{0, 1\}, \{B, R\}, \delta, q_0, R, \phi)$
- Where δ is defined as :
- $\delta(q_0, 0, R) = (q_0, BR)$
 $\delta(q_0, 0, B) = (q_0, BB)$
 $\delta(q_0, 1, B) = (q_1, B)$
 $\delta(q_1, 1, B) = (q_1, B)$
 $\delta(q_1, 0, B) = (q_1, \epsilon)$
 $\delta(q_1, \epsilon, R) = (q_1, \epsilon)$



[3863] – 445

-4-

b) Construct a PDA equivalent to the following CFG 6
 $S \rightarrow 0AA$
 $A \rightarrow 0S/1S/0$

c) Explain how DPDA is different from NPDA. 4

9. a) Construct a turing machine for reversing a string. 8

b) Design a Post machine for the language $L = \{0^n 1^{2n}\}$. 8

OR

10. a) Construct a turing machine for finding 2's complement of a binary number. 8

b) Explain the following terms in relation with turing machine. 8

1) Solvability

2) Semi-solvability and

3) Unsolvability.

www.sppuonline.com

11. a) What is post correspondence problem ? Explain with example. 8

b) What is Halting problem ? Prove that halting problem is undecidable. 8

OR

12. a) Explain the following terms with example.

1) Computational complexity

2) P - class problems

3) NP- class problems

4) Modified PCP problem. 8

b) Define the following terms

1) Recursive language

2) Recursively enumerable language

3) Intractable problems

4) Universal turing machine. 8