UNIVERSITY OF PUNE
[4363]-255
T. E.(Comp. Engineering.)Examination - 2013
THEORY OF COMPUTATION
(2008 Pattern)

[Total No. of Questions:] **[Total No. of Printed Pages :5]**
[Time : 3 Hours] **[Max. Marks : 100]**

Instructions :

*(1) Answers to the **two sections** should be written in **separate answer-books**.*
(2) Assume suitable data, if necessary.

SECTION-I

Q1 a) Define the following with examples: [8]

- Kleen closure
- An alphabet
- Regular expression
- Formal language

b) Design a Moore machine that will read sequences made up of the letters A, E, I, O, U and will give as output the same sequence except in case where I directly follows an E, I will be changed to U. [8]

OR

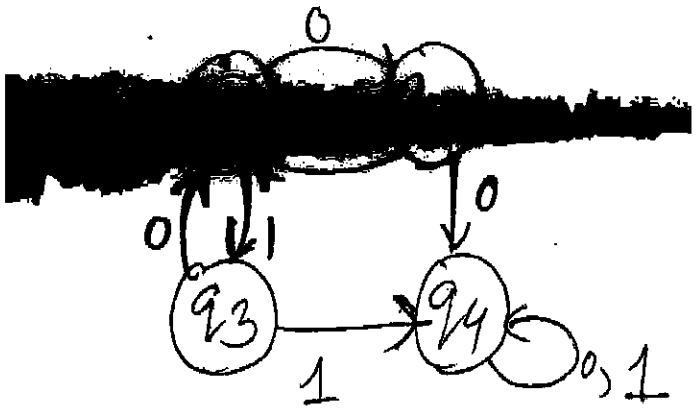
Q2 a) Design a finite automata that reads strings made up of letters in the Word 'CHARIOT' and recognize those strings that contain the word 'CAT' as a substring. [8]

b) Construct DFA equivalent to NFA [8]

	0	1
→P	p,q	p
q	r	r
r	s	-
Ⓢ	s	s

Q3 a) Prove that the FA whose transition graph is as shown below accepts [8]

The set of all strings over the alphabet {0,1} with an equal number Of 0's and 1's, such that each prefix has atmost one more 0 than 1's And atmost one more 1 than 0's.



b) Show that L={a^p / p is prime} is not regular. Make use of pumping lemma and explain the steps in detail. [6]

c) Describe in English language accepted by R.E. $\Rightarrow$ (0+1)* 0 [2]

OR

Q4 a) State and explain in detail the closure properties of regular sets [8]

b) Explain the application of regular expressions in lexical analysis phase of compiler [6]

c) State the pumping lemma for regular sets [2]

Q5 a) Convert the following CFG into CNF(Chomsky Normal Form) [6]

S⟶ABA

A⟶aA|∈

B⟶bB|∈

b) Write a CFG for generating identifiers in higher-level languages such as 'C'. identifiers can be defined by the regular expression(letter). (letter 1 digit)* [4]

c) Obtain a DFA accepting the regular language defined by the following right-linear grammar [8]

S⟶OA|1B

A⟶OC|1A|0

B⟶ 1B|1A|1

C⟶0|0A

OR

Q6 a) Convert the following CFG into GNF(Greibach Normal Form) [8]

S⟶AB

A⟶BS|b

B⟶SA|a

b) Construct a grammar G to represent a language L which is a set of all palindromes over {a,b} [4]

c) Consider the grammar G given by S⟶S+S|S*S|a|b. find the derivation tree for a*b+a*b. is the grammar ambiguous [6]

SECTION-II

Q7 a) Design a PDA to check whether the given expression is a valid Postfix expression. [6]

b) Compare deterministic PDA with non-deterministic PDA [4]

c) Design a PDA for the following CFG G [6]

G= { S ⟶ aAA

A⟶bS

A⟶aS

S⟶a }

OR

Q8 a) The following PDA accepts a language: L=$\{a^n b^m a^n | m,n \geq 1\}$ [12]

Construct equivalent CFG for L such that L(G)=N(A)

A=$(\{q_0 q_1\},\{a,b\},\{a,z_0\},\delta,q_0,z_0,\phi)$

Where δ is given as follows:

$\delta(q_0,q,z_0) = \{(q_0,q,z_0)\}$

$\delta(q_0,q,q) = \{(q_0,aa)\}$

$\delta(q_0,b,a) = \{(q_1,a)\}$

$\delta(q_1,a,a) = \{(q_0,€)\}$

$\delta(q_1,b,a) = \{\delta(q_1,a)\}$

$\delta(q_1,€,z_0) = \{(q_0,€)\}$

b) Compare PDA with FA (finite automata) [4]

Q9 a) Design a turning machine which checks for the language L=$\{a^n b^n\}$ [8]

b) Define Turing machine [2]

c) Design a TM to subtract two unary numbers, the original numbers need not be retained [8]

OR

Q10 a) Design a turing machine to compute 2's complement of a given Binary number. [6]

b) Write short note on universal turing machine along with example [6]

c) Compare NFA, DFA, NPDA, DPDA, turing machines with reference to type of the grammar [6]

Q11 a) Show that if L_1 and L_2 are recursive languages, then $L_1 \cup L_2$ and $L_1 \cap L_2$ are also recursive [8]

b) Explain in detail "Post's Correspondence Problem" with the help of example [8]

OR

Q12 a) Describe in detail Chomsky Hierarchy and context-sensitive languages. [8]

b) Explain in detail the "Halting problem" [6]

c) Define undecidability [2]