**Seat No.: ________** **Enrolment No.___________**

# GUJARAT TECHNOLOGICAL UNIVERSITY

**BE - SEMESTER–VI (OLD) - EXAMINATION – SUMMER 2018**

**Subject Code:160704** **Date:05/05/2018**

**Subject Name:Theory Of Computation**

**Time:10:30 AM to 01:00 PM** **Total Marks: 70**

**Instructions:**

1. **Attempt all questions.**
2. **Make suitable assumptions wherever necessary.**
3. **Figures to the right indicate full marks.**

**Q.1** **(a)** (i) For following regular expression, Draw an FA recognizing the corresponding language **07**

$$(0 + 1)^*(1 + 00)(0 + 1)^*$$

(ii) Write Regular Expression corresponding to each of the following subsets of {0, 1}*

a. The language of all strings containing both 101 and 010 as substrings.
b. The language of all strings in which both the number of 0's and the number of 1's are even.

**(b)** (i) Write Principle of Mathematical Induction. Using Principle of Mathematical Induction, prove that for every n≥0, **05**

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

(ii) Prove by Contradiction that for any sets A, B and C, if A∩B = Ø and C⊆B, then A∩C = Ø. **02**

**Q.2** **(a)** Given the CFG G, Find a CFG G' in Chomsky Normal Form **07**

S → AACD
A → aAb | ^
C → aC | a
D → aDa | bDb | ^

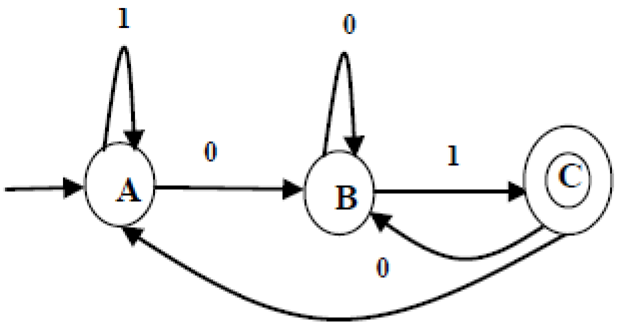
**(b)** Convert the following NFA-^ into its equivalent NFA and FA. **07**
Initial State: A Final State: D

| q | δ(q, ^) | δ(q, 0) | δ(q, 1) |
|---|---|---|---|
| A | {B} | {A} | Ø |
| B | {D} | {C} | Ø |
| C | Ø | Ø | {B} |
| D | Ø | {D} | { Ø } |

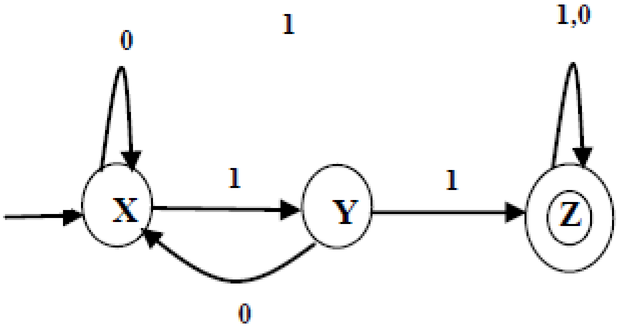
**OR**

**(b)** Let $M_1$ and $M_2$ be the FAs pictured below, recognizing languages $L_1$ and $L_2$ respectively **07**

$M_1$ =




$M_2$ =




Draw the FAs recognizing the following languages:

- $L_1 \cap L_2$
- $L_2 - L_1$

**Q.3** **(a)** (i) Write a CFG for solving simple ( & parenthesized) expression, such as + and *. **07**

(ii) Also write CFG fir regular expression r = (a + b) (a + b + 0 + 1)*
Use CFG defined for part(i).

(iii) Derive the string (which is defined in part(ii)) a * (a + b00) by applying left most derivation and right most derivation.

**(b)** (i) Show that the function f(x, y) = x*y is primitive recursive. **03**

(ii) Find the transitive closure and the symmetric closure of the relation **02**
{ (1,2), (2,3), (3,4), (5,4) }

(iii) Give recursive definition for the language L which is the set of all integers (positive and negative) divisible by 7. **02**

**OR**

**Q.3** **(a)** Define Context Free Grammar. Design a CFG for the language **07**

$L = \{ a^i b^j c^k \mid i \neq j + k \}$

**(b)** (i) Define bijection function. Explain Compositions and Inverses of Functions. **07**

(ii) Define Primitive Recursive Function. Show that Addition function of two positive integers is primitive recursive.

**Q.4** **(a)** (i) Design a PDA to recognize the language generated by the following grammar: **04**

S → 0AB
A → 1A | 1
B → 0B | 1A | 0

Show the acceptance of the input string string “011100“ by this PDA.

(ii) Prove that the language
L = {ww|w is in (0 + 1)*} is not a CFL. **03**

**(b)** (i) Explain Universal Turing Machine. **04**

(ii) Prove the theorem: “A language is recursive if and only if both it and its complement are recursively enumerable.“ **03**

**OR**

**Q.4** **(a)** (i) Prove that $L = \{a^n b^n c^n \mid n \geq 0 \}$ is not a CFL using pumping lemma. **03**

(ii) Consider following PDA machine M = ({p, q}, {0,1}, (x, z}, δ, q, Z) where δ is given by **04**

δ(q, 1, z) = (q, xz)
δ(q, 1, x) = (q, xx)
δ(q, ^, x) = (q,^)
δ(q, 0, x) = (p, x)
δ(p, 1, x) = (p, ε)
δ(p, 0, z) = (q, z)

Construct Equivalent CFG.

**(b)** Write Short Note on following: **07**

(i) Halting Problem
(ii) Explain P and NP Completeness

**Q.5** **(a)** Design DPDA for the language L that accepts strings with more a's than b's. Trace String "abbabaa". **07**

**(b)** Design a Turing Machine that creates a copy of its input string. Trace String "baa". **07**

**OR**

**Q.5** **(a)** Construct pushdown automata for the following language: **07**

L = {the set of strings over alphabet {a, b} with exactly twice as many a's and b's }

Trace string "abaabbaaa".

**(b)** Design a Turing Machine which recognizes words of the form $a^n b^n c^n \mid n \geq 1$. Trace string "aabbcc". **07**

*************