# **Theory of Automata (I)**

**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES**

## Final Examination, Fall 2012

## Section A,B,C,D,E

**Total Marks: 100 (20 objective+80 subjective)**

**Time: 3 Hours**

**Subjective Part: (2.5 hrs)**

**Q1.** **Regular languages**

**(i)** [5 points]

Develop DFA for the language over the alphabet {a, b, c} that accepts strings whose last two characters are the reverse of first two characters.

**(ii)** [10 points]

Use the state elimination method to convert the following DFA to a regular expression. You must show all the intermediate steps.



**Q2.** **Context Free Languages**

**(i)** [10 marks]

Build CFG for the following language: {ai bj ck | *j* not equal to *i* + *k*}

**(ii)** [5 marks]

Is the language given below a CFL? If not prove via Pumping Lemma.

L = {www| w belongs to {0,1}\*}

**(iii)** [10 marks]

**Constructing an LL (1) parser.**

Consider the following CFG over ∑ = {, , d}



S 🡪 SS | S S | d



Convert it into LL (1) Grammar and then build its LL (1) parser.

**Q3.** **Turing Machine**

**Note:**

* Give short description of your TM in few points.
* In case you are using any blocks (composite TM) to design this Turing machine, you need to specify the internal design of those blocks as well.
* **For Section A,B,C,D design TM for the parts below & for Section E design Deterministic TM for the parts below.**

**(i)**  [10 marks]

Design a generic 1-Tape Turing machine “EQUAL” that will accept the input if it contains an equal number of 0’s and 1’s. Input is placed on the tape. The machine should accept 01111000, 11110000, 10110100…..

**(ii)** [15 marks]

Design a generic 3-Tape Turing machine for Div (am, an) function where m and n > 0. You need to divide string am with string an . Initially Tape-1 contains am and then an  separated by # and tape 2 & 3 are blank. After division, tape-3 should contain the quotient.

**Sample Run:**

Input: (on tape-1 initially)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | a | a | a | a | a | # | a | a | # |  |

Output: (on tape-3 after execution)

|  |  |  |  |
| --- | --- | --- | --- |
| # | a | a |  |

**(iii)** [15 marks]

Design a standard 2-tape Turing machine with alphabet set Σ = {a, b, c} that sorts the symbols in the input string in the lexicographic order such that in the output string all a's appear before the b's and c's, and all b's appear before the c's (For example, on input bbcacb, the output would be abbbcc.).

You can assume that the input is on tape 1. Your output should be on tape 2 & input on tape 1 when the TM halts.

**Sample Run:**

**Input**(on tape-1 initially)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ∆ | b | b | c | a | c | b | ∆ | ∆ | ∆ |

Output: (on tape-2 after execution)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ∆ | a | b | b | b | c | c | ∆ | ∆ | ∆ |