**COMPUTER SCIENCE DEPARTMENT**

**MILITARY COLLEGE OF SIGNALS, NUST**

**MID-TERM EXAMINATION**

**THEORY OF AUTOMATA AND FORMAL LANGUAGES**

**BESE – 14 A & B (SOFTWARE ENGINEERING TRACK)**

**INSTRUCTOR TIME: 90 MINUTES**

**UMAR MAHMUD MAX MARKS: 30**

***Note***

* *Attempt all questions and all parts.*
* *Provide sound reasoning and supporting examples where necessary.*
* *Remember ‘A picture is worth a thousand words’.*
* *If required, state your assumptions clearly.*
* *Do not give false arguments. This may result in a negative impact.*
* *There are* ***2*** *pages of the examination.*

# QUESTION 1 - INTRODUCTION

1. What is a finite language? Give two examples? (1+1)
2. Differentiate among alphabet and strings with examples. (1+1)

# QUESTION 2 – FINITE AUTOMATA

1. What is a Trap state? Give example. (1+1)
2. What are the advantages of NFA over DFA? Explain with examples. (1+1)
3. Construct a DFA in that accepts all strings that start with exactly two ‘α’ and does not have more than 5 ‘β’ in the string. (4)
4. For the transition matrix construct appropriate NFA. Also convert the resulting NFA to the DFA using the algorithm where and. (5)

Table 1:Transition Table

|  |  |  |  |
| --- | --- | --- | --- |
|  | **a** | **b** | **c** |
| **q0** | q0 | q1 | q2 |
| **q1** | - | q0 | q3 |
| **q2** | q4 , q5 | q1 , q3 | - |
| **q3** | q4, q5, | q0, q1, q4 | q5 |
| **q4** | q5, q2 | - | q4 |
| **q5** | q5 | q4 | q3, q5 |

1. What is the language generated by the NFA shown in Figure 1. (2)

Figure 1: NFA

# QUESTION 3 – REGULAR EXPRESSIONS

1. Define Union and Kleene Star operation on strings with examples. (2+1)
2. What is a generalized NFA? Give example. (1+1)
3. Write the regular expression for the following languages (4)
   1. All strings of a’s and b’s beginning with exactly one b.
   2. All strings of a’s and b’s ending in atleast one b.
   3. All strings of a’s and b’s that contain atleast one a.
   4. All strings of a’s and b’s that contain atleast two b’s.
4. For, draw appropriate finite automaton for the regular expression. (2)

# THE END