Tribhuwan University
Institute of Science and Technology
2075

Bachelor Level / fourth-semester / Science Full marks: 80 **Computer Science and Information Technology(CSC257)** Pass marks: 24 (Theory of Computation) Time: 3 hours Candidates are required to give their answers in their own words as far as practicable.
The figures in the margin indicate full marks.

Attempt all the questions.

Group A (8x4=32)

1. How can you represent a finite Automata? Explain.

2. Construct a DFA that accepts the strings over the alphabet {0,1} with odd number of 0's and even number of

1's. 3. Define the ε-closure of a state of ε-NFA with an example.

4. Write regular expressions for the following regular languages.

a) The set of strings over an alphabet {a, b} containing at least one 'a' and at least one 'b'.

b) The set of strings over {0, 1} whose 5th symbol from the right end is 1.

5. Given the following grammar

$$E \rightarrow E + T \mid E - T \mid T$$
$$T \rightarrow T^* F \mid T/F \mid F$$
$$F \rightarrow (E) \mid id$$

Remove the immediate left recursion from the grammar.

6. How can you convert a CFG into equivalent PDA? Explain with examples.

7. What is a Turing Machine? Give formal definition. How does it differ from FA?

8. What do you mean by tractable and intractable problems? Are intractable problems solvable by the Turing

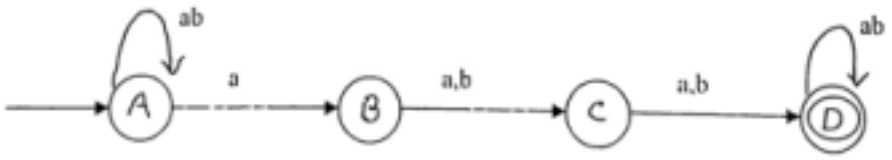
machine?

Group B (6x8=48)

9. Convert the following regular expression into ε-NFA.

a) 01* b) (0+1)01* c) 00+(0+1)*100*

10. Convert the following NFA into equivalent DFA using subset construction and also show the transition diagram for this DFA.



11. Convert the following grammar into Chomsky Normal Form.

S→ ASB|ε

A → aAS|bAS|a

B → SbS|A|CS|bb

12. Construct a Push Down Automata that accepts all the strings from the alphabet {0, 1} with equal numbers of 0 and 1. Show that 0110 is accepted by this PDA and 01101 is not.

13. How can you show that the one tape Turing machine and multi-tape Turing machine are equivalent? Explain in detail.

14. Explain the term Turing acceptable and Turing decidable. Show that if L is a recursive language then complement of L is also recursive.