**Homework # 2**

**Theory of Automata and Formal Languages**

|  |  |
| --- | --- |
| Total Marks: 75 | Course Code: CS-3131 |

**Note: All questions carry equal marks.**

Q1: Eliminate the transitions from the following NFA.

|  |
| --- |
|  |

Q2: Define closure of a set of states along with an example, in case of NFA-.

Q3: What is Kleene’s Theorem? Explain it with examples.

Q4: Build a NFA- for a regular expression ((aa + b)∗(aba)∗bab)∗ using the concepts of Kleene’s theorem.

Q5: What are the pre-processing rules of State Elimination Algorithm. Explain all with examples.

Q6: Find a regular expression for the following FA using State Elimination Method.

|  |
| --- |
|  |

Q7: Apply the concept of equivalence of states via DFA minimization method on the following FA.

|  |
| --- |
|  |

Q8: Discuss the Pumping Lemma and its properties.

Q9: Prove that the language is not a regular language.

Q10: Is the following CFG ambiguous? Check your answers for the given string “a + a ∗ a ”. Support your answers further through the examples of derivations and their respective parse trees as well. Finally, remove the ambiguity of the grammar

S → a | S + S | S ∗ S | (S)

Q11: Discuss the pre-simplification rules for CFG to convert it into CNF.

Q12: Convert the following CFG into CNF.

|  |
| --- |
|  |

Q13: In the following CNF grammar, show that the string ‘baaba’ is the member of this given grammar using the CYK algorithm.

|  |
| --- |
|  |

Q14: Design a PDA for the following grammar S🡪 aSa | bSb | , and also show the transition table with entries of at least stack and tape for the string ‘abba’.

Q15: Design a Turing machine for computing the reverse of a given string if Σ=(a,b).