| **STUDENT PORTFOLIO** | |
| --- | --- |
| Insert Photo | **Name:H.MADAR HUSSAIN KHAN**  **Register Number:RA2011030010109**  **Mail ID:mk3709@srmist.edu.in**  **Department: NWC**  **Specialization: CSE CYBER SECURITY**  **Semester:5TH** |
| **Subject Title: 18CSC301J Formal Language & Automata**  **Handled By: LAVANYA** | |
| **Assignment – Work Sheet (Unit 1, 2, 3, 4, & 5)**  **(Write about the assignment questions and how you solved differently)**        DESCRIPTIVE QUESTIONS   1. Show that 22n-1 is divisible by 3 using the principles of mathematical induction. 2. Prove that if for an integer a, a2 is divisible by 3, then a is divisible by 3 using the proof by contradiction. 3. For any two integers a and b, (a+b) is odd if and only if exactly one of the integers a or b is odd. Prove the above statement. 4. notShow by counter example the given statement P is always true.   P = 2n2-16n+31 is always positive for all of n.   1. Prove using mathematical induction for n>=5, 2n>n2. 2. Prove that the sum of n squares can be found as follows   12+22+32+...+n2=n(n+1)(2n+1)/6        **UNIT 2 WORKSHEET**  **(1)**   1. Manish has to travel back to this home every day from college. For, that he can use 2 paths, path A and path B. He can use any path, but he has to go for 10 days, every day construct a CFG for the language of L= an b2n ,where n>=1, that is If he takes the path A on day one , he has to go by path B for the next two days.   For the above scenario, take some example route and perform left most and right most derivation, check whether the constructed grammar in ambiguous or not.   1. Construct a grammar for the desktop calculator App to perform addition, subtraction, multiplication and division operation restricted to the integers restricted to the integers {2,3,4} and check whether the grammar is ambiguous or not using Parse tree. 2. Construct a grammar representing syntactic structure {{IF, THEN}, {IF, THEN, ELSE}, {IF, THEN, ELSE, IF}} statements in C programming. Consider an example and check whether the constructed grammar is ambiguous or not.         **UNIT 3 WORKSHEET 1**   1. Give pushdown automata that recognize the following languages. Give both a drawing and 6-tuple specification for each PDA.   A = { w ∈ {0, 1} ∗ | w contains at least three 1s }  **B = { w ∈ {0, 1} ∗ | w = wR and the length of w is odd }**  **C = { w ∈ {0, 1} ∗ | w = wR }**  **D = { a i b j c k | i, j, k ≥ 0, and i = j or j = k }**  **E = { a i b j c k | i, j, k ≥ 0 and i + j = k }**  **F = { a 2n b 3n | n ≥ 0 }**  **L = { a i b j c k | i, j, k ≥ 0 and i + k = j }**  **h=∅, with Σ = {0, 1}**  (i) The language H of strings of properly balanced left and right brackets: every left bracket can be paired with a unique subsequent right bracket, and every right bracket can be paired with a unique preceding left bracket. Moreover, the string between any such pair has the same property. For example, [ ] [ [ [ ] [ ] ] [ ] ] ∈ A.   1. (a) Use the languages A = { a mb n c n | m, n ≥ 0 } and B = { a n b n c m | m, n ≥ 0 } together with Example 2.36 of the textbook to show that the class of context-free languages is not closed under intersection.   (b) Use part (a) and DeMorgan’s law (Theorem 0.20 of the textbook) to show that the class of context-free languages is not closed under complementation.   1. Consider the following CFG G = (V, Σ, R, S), where V = {S, T, X}, Σ = {a, b}, the start variable is S, and the rules R are S → aT Xb T → XT S | ε X → a | b Convert G to an equivalent PDA using the procedure given in Lemma 2.21.       **UNIT 4 WORKSHEET 2**  **MCQ -5**   1. The difference between a read-only Turing machine and a two-way finite state machine is 2. Head Movement 3. Finite Control 4. Storage Capacity 5. Power 6. Which of the following is true for two stack Turing machines? a) one read only input b) two storage tapes c) one read only input & two storage tapes d) two read only input & two storage tapes 7. If instead of moving left or right on seeing an input, the head could also stay at one position without moving anywhere is called as \_\_\_\_\_\_\_\_ 8. Turing Machine with Fixed Tape 9. Turing Machine with Stay option 10. Turing Machine with Semi-infinite tape 11. Offline Turing machine 12. In standard Turing machine the input symbol can be changed to blank, but if we remove this facility of changing the input symbol to blank then such type of Turing machine is called as \_\_\_\_\_\_\_\_\_\_\_\_\_ 13. Non erasing Turing Machine 14. Jumping Turing Machine 15. Always writing Turing Machine 16. Offline Turing machine 17. A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is one whose tape alphabet consists of exactly two symbols. 18. Alphabet based Turing Machine 19. Binary Turing Machine 20. Count based Turing Machine 21. Symbols based Turing Machine   **Descriptive Question:**   1. Whether it is possible to increase the number of languages accepted by performing some modifications in Standard Turing Machine? If Yes, Justify the ways of modifications.   **Scenario Based Question:**   1. A Turing machine with doubly infinite tape is similar to an ordinary Turing machine, but its tape is infinite to the left as well as to the right. The tape is initially filled with blanks except for the portion that contains the input. Computation is defined as usual except that the head never encounters an end to the tape as it moves leftward. Show that this type of Turing machine recognizes the class of Turing-recognizable languages.   **Worksheet Question:**   1. Design a Multi tape Turing Machine for L= anbncn | |
| **Assignment**  **(what is the most interesting part in the assignment)**  **THE MOST IMPORTANT PART IS A GOOD PRACTISE SKILL AND COULD DEVELOP SOME IDEA ABOUT THIS AUTOMATA ,HELPED TO GET A ABSTRACT IDEA OF THE FORMATION AND RULES** | |
| **Hacker Rank Achievements** | |
| **Any other**  **(Write if you registered or practise apart from Codechef (ex. Hackerrank, Leetcode etc.)**  [**https://www.hackerrank.com/mk3709**](https://www.hackerrank.com/mk3709)  **https://github.com/mk3709/regex-** | |

|  | **Signature** | | | | | | | | | | | | |  | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| **Note: Enclose the assignment and relevant certificates along with the profile** | | | | | | | | | | | | | | | |