National University of Computer & Emerging Sciences, Karachi

Spring-2025 CS-Department

Assignment 2

Issue Date: **Due date:** **Total Marks 70**

Course Code: CS301	Course Name: Theory of Automata

Instructions:

- **This is hand written assignment (Docs work not allowed).**
- **Write the correct question number with part number (e.g : Q1 part a).**
- **You can use A4 size paper for solving the assignment.**
- **Attempt all the parts of Question to get full marks (no shortcuts, proper workflow as taught in the course).**

You must take clear snaps of your own handwritten assignment, make a proper pdf file and then submit it on the Google Classroom Submission before the deadline.

Question 1 Kleen's Theorem **Points:15**

1. Perform concatenation of DFA mentioned in Figure 3 and DFA mentioned in Figure 4 using Kleen's Theorem.
2. Perform Kleen's Star Closure of DFA mentioned in Figure 3 .
3. Perform Kleen's Star Closure on the Given DFA in Figure 5.

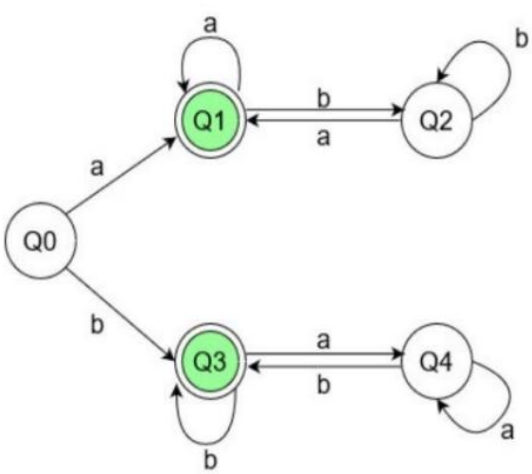
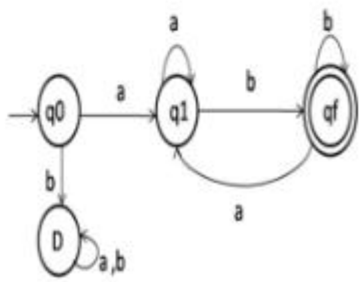


Figure 5



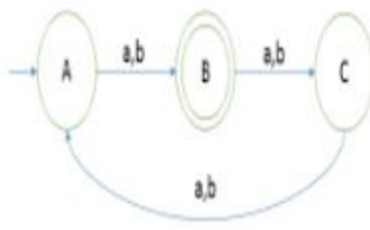


Figure 3 Figure 4

Question 2. Pumping lemma **Points:15**

Apply Pumping Lemma on languages **L_1** ,**L_2** and **L_3** to prove that these are not regular languages.

(i)L_1= *{0^n 1^m : n≤m+3}*

(ii) L_2= *{01(1100)^n 110(10)^n :n≥0}*

(iii) L_3 = *{a^i b^j c^k : i>=j>=k>=1}*

Question 3. Minimization of DFA **Points:10**

Minimize the given DFA using any method. i.e Partition or Myhill-Nerode Theorem

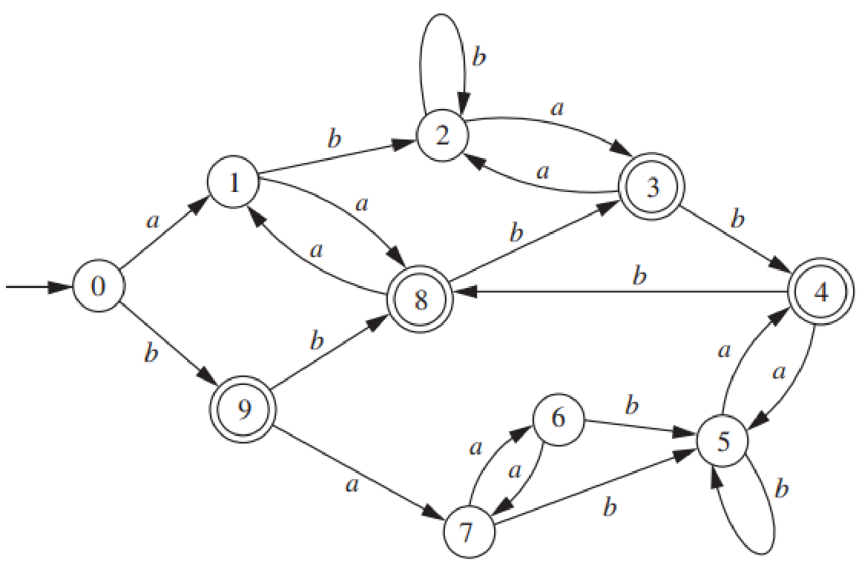


Figure 2

Question 4. Mealy to Moore & Moore to Mealy Conversion

(a) Consider the given mealy machine convert it into equivalent moore machine

Points:5

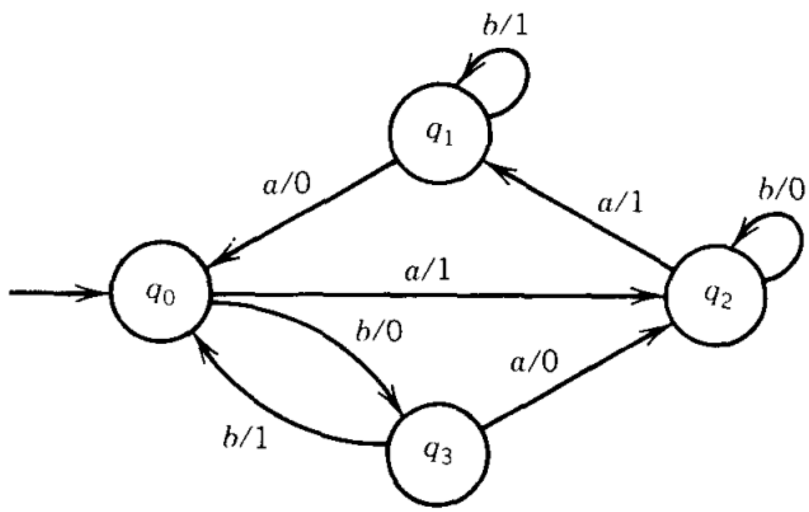


Figure 1

(b) . Draw the Mealy machine for the following scenario. **Points:5**

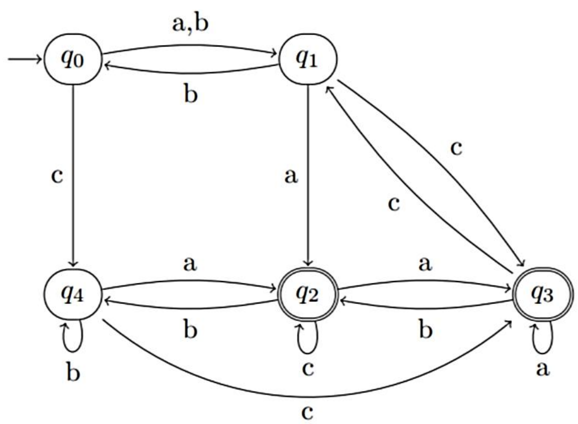
You are going to develop a simple shooting game. There is one hero in a battlefield with few enemies and other characters. There are three states in the game which are WANDER, EVADE and ATTACK. When the hero is wandering the field and suddenly encounters enemies while he is not in vulnerable situation, he will go into ATTACK stage by shooting the enemies. However, if while wandering the field and the hero suddenly encounters enemies and he is in vulnerable situation, he will shoot the enemies and goes into EVADE stage. While in the ATTACK stage, if the hero encounters enemies and he is not vulnerable, he will remain in that stage and continue shooting. But if he encounters enemies and he is vulnerable, he will shoot and go into EVADE stage. While in EVADE stage, if the hero encounters enemies and he is not vulnerable, he will go into the ATTACK stage and shoot. But if he encounters enemies and he is vulnerable, he will remain in that stage and continue shooting. When the hero encounters characters that are not his enemies, he will not shoot. If he is in the WANDER stage at that time, he will continue wandering the field. If he is in the ATTACK or EVADE stages, he will switch into the WANDER stage. The inputs and outputs are given in Table 1 below. Construct a finite state diagram to model the game.

States	Input	Output
Evade (**E**)	Not Enemies (**NE**)	Shoot (**S**)
Wander (**W**)	Enemies and Not Vulnerable (**ENV**)	Not Shoot (**NS**)
Attack (**A**)	Enemies and Vulnerable (**EV**)	

Table 1

Question 5 Properties of Regular Language

(a) Construct a Regular Grammar for the following DFA: **Points: 5**



Requirements:

1. Identify the set of states for the DFA.
2. Define transitions based on the given grammar.
3. Clearly indicate the start and accepting states.
4. Draw a state diagram representing the DFA.

(b) Convert the following Regular Grammar into a Regular Expression. Show your work step by step. **Points: 5**

$$A \rightarrow aB \mid bC \mid \lambda$$

$$B \rightarrow aB \mid bB \mid \lambda$$

$$C \rightarrow aC \mid b$$

(c) Give Regular Grammar for *{b, abc, aabcc, aaabccc, … $a^n b c^n$ …}* **Points: 5**

(d) Properties of Regular Grammar **Points: 5**

(i) Prove that the language defined by A* for the grammar:

$$A \rightarrow aA \mid \epsilon$$

generates all strings of **a's**.

(ii) Construct a regular grammar for **concatenation** of two languages:

- L1= (a+b)*a
- L2= b(a+b)*

Hint: Define separate grammars for L_1 and L_2 and then derive a new grammar that generates concatenation.

(iii) Given two supposed regular grammars G_1 and G_2, prove that the **substitution** operation preserves regularity.

(iv) Prove or Disprove:
"If a language is defined by a **left-linear** grammar, then it must be regular."