National University of Computer & Emerging Sciences, Karachi
Spring-2022 School of Computing
Final Examination
6th June, 2022, 8:30 am to 11:30 am

Course Code: CS3005	Course Name: Theory of Automata
Instructor Name(s): Mr. Musawar Ali, Ms. Bakhtawar Abbasi	
Student Roll No:	Section:

Instructions:

- Return the question paper.
- Attempting of the question in the given order is highly encouraged.
- Read each question completely before answering it. There are **9 questions on 4pages.**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper. **Give short answers where possible.**
- **Show full steps and provide appropriate reasons wherever possible to get full credit.**
- **Short answers should be short and must not be in story form to get full credit.**

Time: 180 minutes. **Max Marks: 100 points**

Question 1: Miscellaneous Short Questions (CLO 2) **(2x5=10) Points**

- **a.** Differentiate between the semantic and syntactic languages.
- **b.** Draw the Chomsky's hierarchy **Venn diagram**, with appropriate FSMs and grammars associated with the languages.
- **c.** Enlist the properties of RLs, for which RLs remain closed.
- **d.** Enlist the properties of CFLs, for which CFLs are not closed.
- **e.** Draw two PDAs for **a*** and **b*** perform union on the PDAs. Follow the method of properties of regular languages used for union.

Question 2: Kleens Theorem (CLO 2) **(2x5=10) Points**

- **a.** Minimize the **DFA1** using any method of your own choice.
- **b.** Perform union of **DFA1** and **DFA2** using Kleen's Theorem.
- **c.** Perform intersection of **DFA1** and **DFA2** using Kleen's Theorem.
- **d.** Perform concatenation of **DFA1** and **DFA2** to make **DFA2.DFA1** using Kleen's Theorem.
  Hint: $a.\phi = \phi \quad and \quad \phi.a = \phi$
- **e.** Consider **A** as the accepting state in **DFA 1**, and perform Kleen Star Closure on the DFA.

Note: Short and logical answers are required in this question. Don't go for long methods also no need to draw machines.

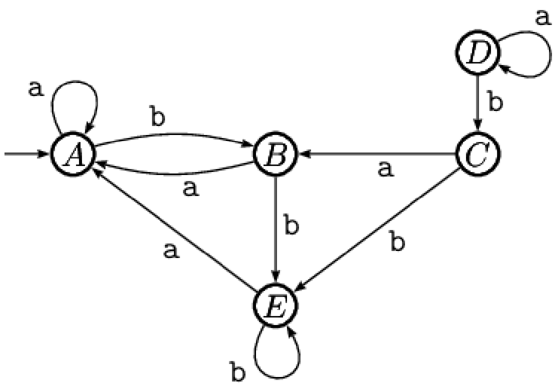


Figure 1 DFA1

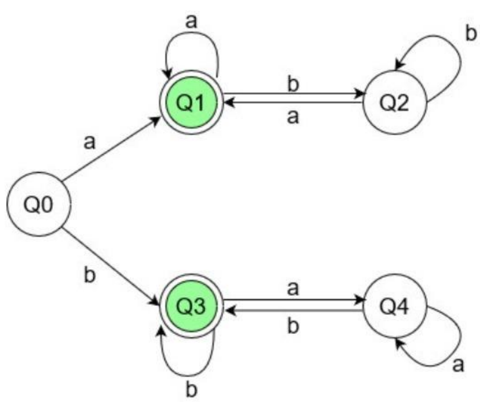


Figure 2 DFA 2

Question 3: Regular Expressions and DFA (CLO 1) **(2x5=10) Points**

a. Give RE for L_1 = Language of words having a's on even indices, where the index starts from 1. Defined over the input alphabet $\Sigma = \{a, b\}$. Example words = {**λ, aa, ba, aaba, baba,..**}.
b. Give RE for L_2 = Language of words having b at the center and having odd a's at right and left of b. Defined over the input alphabet $\Sigma = \{a, b\}$.
c. Draw a DFA for L_3 = Even odd language. Defined over input alphabet $\Sigma = \{a, b\}$.
d. Draw a DFA for $L_4 = \{a^i b^j c^k d^l | i, j, k, l \geq 0 | \ k < j\}$. Defined over input alphabet $\Sigma = \{a, b\}$.
e. Draw a DFA for the language accepting strings such that each **'0'** is immediately preceded and followed by **'1'**. Defined over input alphabets $\Sigma = \{0, 1\}$.

Question 4: Mealy Machine, NFA to DFA and GTG (CLO 4) **(6+2+2=10) Points**

a. Draw the Mealy machine for the following scenario.

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 **ATTACK** stage, if the hero encounters enemies and he is not vulnerable, he will remain in that stage and continues shooting. But if he encounters enemies and he is vulnerable, he will shoot and goes into **EVADE** stage. While in **EVADE** stage, if the hero encounters enemies and he is not vulnerable, he will go into **ATTACK** stage and shoots. But if he encounters enemies and he is vulnerable, he will remain in that stage and continues 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 **ATTACK** or **EVADE** stages, he will switch into **WANDER** stage. The inputs and outputs are given in the 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**)	

b. Convert the **NFA1** to DFA. Give the transition table of NFA 1 and convert it into DFA, give your conclusion.
c. Determine the RE of the **NFA2** using GTG and state elimination method.

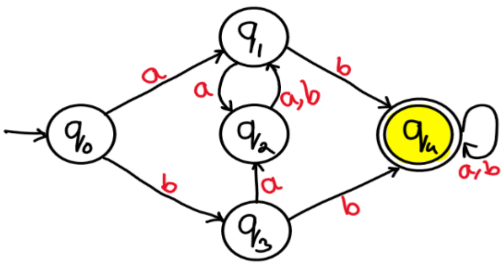


Figure 3 NFA 1

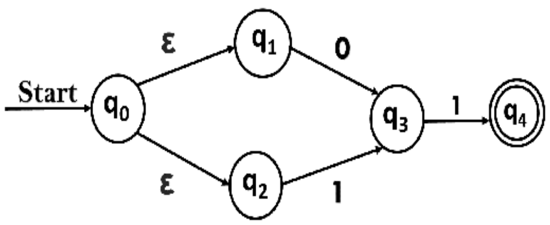


Figure 4 NFA 2

Question 5: CFG and RG (CLO 2) **(2+2+2+1+2+1=10) Points**

- **a.** Design the CFG/RG for $\boldsymbol{L_4} = \{\text{even-even language}\}$. Defined over alphabet $\boldsymbol{\Sigma} = \{\boldsymbol{a}, \boldsymbol{b}\}$.
- **b.** Design the CFG/RG for the language $\boldsymbol{L_1} = \{a^n b a^n | \boldsymbol{n} \geq \boldsymbol{0} | n\ is\ odd\}$. Defined over alphabet$\boldsymbol{\Sigma} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$.
- **c.** Design the CFG/RG for the language $\boldsymbol{L_2} = \{a^n b^m c^o | \boldsymbol{o} \leq \boldsymbol{m} |\ m, n, o \geq 1\}$. Defined over alphabet $\boldsymbol{\Sigma} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$.
- **d.** Design the CFG/RG for the language $\boldsymbol{L_3} = \{a^n | n\ is\ prime\}$. Defined over $\boldsymbol{\Sigma}\ = \{\mathbf{a}\}$
- **e.** Find the union of CFGs of languages $\boldsymbol{L_1}\ and\ \boldsymbol{L_2}$.
- **f.** Draw the derivation tree for the words '**aaabaaa**' and '**aaa**' from $\boldsymbol{L_1}$and $\boldsymbol{L_3}$ respectively.

Question 6: Ambiguity and CFG to PDA (CLO 2) **(2x5=10) Points**

Consider the following CFG for non-empty language:

S→AxA | By | zC

A→B | a

B →C | b

C→SS | c | λ | ZBC

Z→aBZ | BC |Z

- **a.** Remove useless productions from the given CFG, if there are any.
- **b.** Check for the ambiguity in the given CFG for the string '**axyy**'.
- **c.** Convert the given CFG into PDA.
- **d.** Give the instantaneous description of resultant PDA.
- **e.** Trace the input string '**yyy**' on stack using the resultant PDA.

Question 7: Simplification of CFG and CNF (CLO 2) **(2x5=10) Points**

Consider the following CFG for non-empty language:

S→AB

A→aAb| λ

B→bBc| λ | D

C→cC| λ

D→aDc

- **a.** Simplify the given CFG.
- **b.** Covert the resultant CFG into CNF.
- **c.** Find the closure of CFG in the simplified form. Just mention the production used for closure.
- **d.** Does the CNF remove ambiguity in the CFGs? Justify?
- **e.** Why do we simplify the CFGs, keeping in the view that usually the number of productions increase after simplification?

Note: For part **a** and **b** show the steps clearly in rectangle boxes, highlight your simplified CFG and final CFG (converted into CNF) with appropriate tags to get full credit.

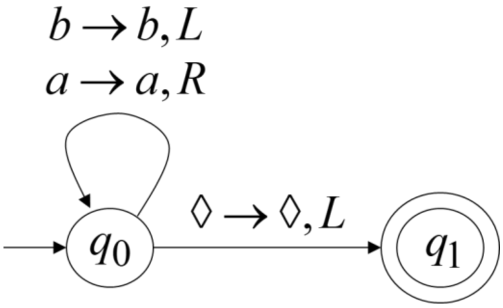
Question 8: **PDA (CLO 5)** **(5+5=10) Points**

Design PDAs for following languages.

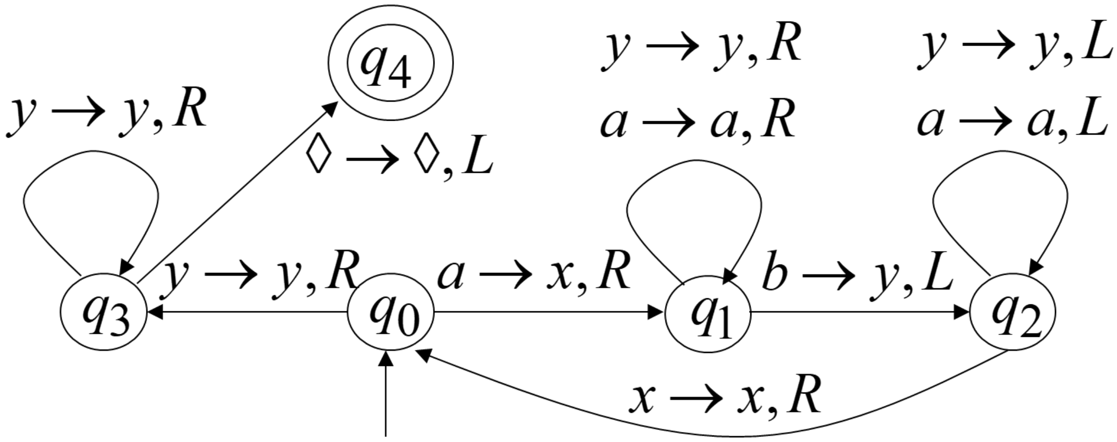
- **a.** $L_3 = \{a^n b^{2m} c^{3n} d^m \mid n, m > -1\}$. Defined over the alphabet $\Sigma = \{a, b, c, d\}$.
- **b.** $L4 = \{a^n b^m \mid n \leq m + 3 \mid n, m \geq 0\}$. Defined over alphabet $\Sigma = \{a, b, c\}$.

Question 9: **Turing Machine (CLO 5)** **(2+2+2+2+2+2+4+4=20) Points**

- **a.** Jack designed following Turing Machine for **(a+b)*** and he put the word '**aabba**' on input tape and gets stuck in the way due to a problem in the machine. You need to identify the problem and provide a solution to problem by drawing the input tape which contains the word '**aabba**' and redraw the Turing machine which is error free.



- **b.** Differentiate between '**accept and halt**' and '**reject and halt**'. Give the short answer.
- **c.** Elaborate the concept of determinism in Turing Machine. In the given machine q0 has no outgoing transitions of b. Can we call this machine deterministic? The input alphabet of machine is $\Sigma = \{a, b\}$.



- **d.** Perform closure on given machine.
- **e.** Can the Turing machine have multiple final/accepting states?
- **f.** Design the Turing machine for **odd-odd** language defined over alphabet **Σ= {a, b}**.
- **g.** Draw the Turing Machine for language $L_1 = \{a^n d b^m \mid n, m \geq 0\}$. Defined over alphabet $\Sigma = \{a, b, d\}$.
- **h.** Draw the Turing Machine for language $L_2 = \{a^n b^m c^n d^m \mid n \geq 1 \mid m \geq 0\}$. Defined over alphabet $\Sigma = \{a, b, c, d\}$.

Good Luck