National University of Computer & Emerging Sciences, Karachi
Spring-2022 CS-Department
Mid-I Exam
11th of March 2022, 10:00 am – 11:00 am

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

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 **5 questions on 3 pages.**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.

Time: 60 minutes. **Max Marks: 40 points**

Question 1: Languages and Alphabets **(2+2+2+2) Points**

Consider the following alphabets and list of the words and answer below mentioned questions:

Σ={Aa,D,Bc,Cd}

Marks Distribution

0.75 if 1 is correct

1.5 if 2 are correct

2 if are correct

(i) **AaBcCd**

(ii) **AaDBcCd**

(iii) **AaDAa**

a. Check the validity of mentioned alphabet based on the listed words.

All are valid based on alphabet, it has no same prefixes.

b. Take the reverse the of mentioned words.

CdBcAa

CdBcDAa

AaDAa.

c. Give the cardinality of each word.

AaBcCd →3

AaDBcCd →4

AaDAa→3

d. Let the alphabet be $\Sigma = \{\, a,\ b,\ c\}$ and we suppose that language L_1 of words in $(a^n b^n + c^n)$. List the words in L_1 which would be included in L^2 and L^3.

Marks Distribution

2 if correct, 0.5 if wrong

$$L^2 = a^n b^n a^n b^n + a^n b^n c^n + c^n a^n b^n + c^n c^n$$

$$L^3 = (a^n b^n a^n b^n + a^n b^n c^n + c^n a^n b^n + c^n c^n)(a^n b^n + c^n)$$

$=a^n b^n a^n b^n a^n b^n + a^n b^n c^n a^n b^n + c^n a^n b^n a^n b^n + c^n c^n a^n b^n + a^n b^n a^n b^n c^n + a^n b^n c^n\, c^n + c^n a^n b^n\, c^n + c^n c^n\, c^n$

Question 2: Deterministic Finite Automata **(3+3) Points**

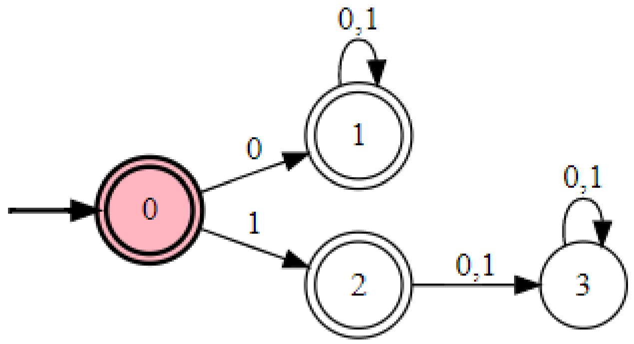
Marks Distribution

1 if wrong attempt

2 if little mistake

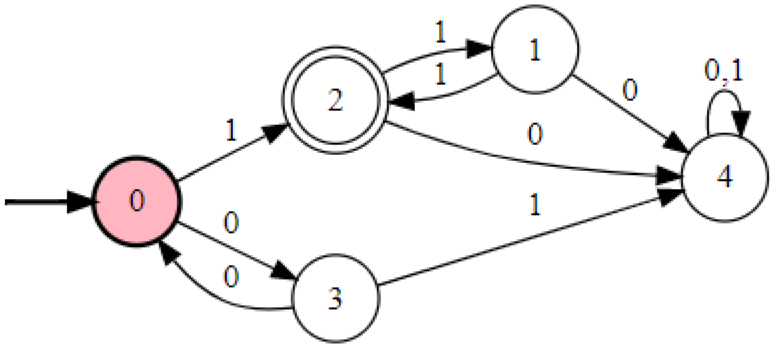
3 if correct

a. Design the DFA for Regular Expression $0 + 1 + \lambda + 00^*1^*(0+1)^*$. Defined over alphabet $\Sigma\{0,1\}$.



b.

c. Design the DFA for Regular Expression (00)* (11)*1. Defined over alphabet Σ{0,1}.



Question 3: **Regular Expressions** **(3+3) Points**

Marks Distribution

1 if wrong attempt

2 if little mistake

3 if correct

a. The language of the words which does not have 0,1 as the third last letter. Defined over alphabet $\Sigma = \{0,1\}$.

$$(0+1+\lambda)(0+1+\lambda)$$

Or $(0+1)(0+1)+1+0+\lambda$
or$(0+1+\lambda)(0+1)+\lambda$

00

b. The set of strings of even length strings but ends with aa. Defined over alphabet $\Sigma\{a,b\}$.

$$\left((a+b)(a+b)\right)^{*}aa$$

Question 4: **Generalized Transition Graph** **(5 Points)**

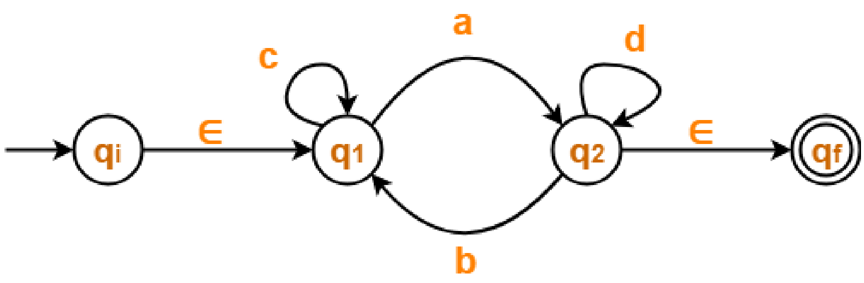
Find the regular expression of given ϵ-NFA using state elimination method.

Marks Distribution

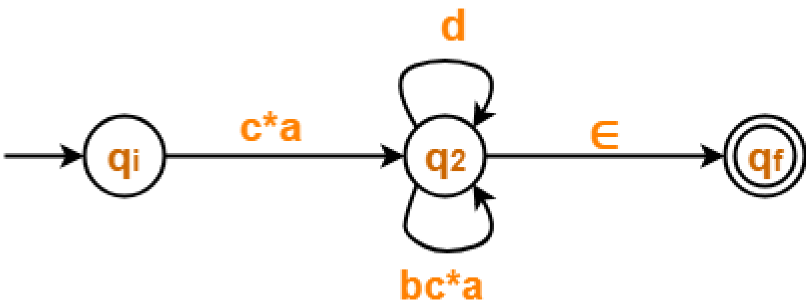
1 if something irrelevant

3 if eliminates state q2

5 if correct



Eliminating state q_1, we get-



Now, let us eliminate state q_2.

- There is a path going from state q_i to state q_f via state q_2 .
- So, after eliminating state q_2, we put a direct path from state q_i to state q_f having cost c*a(d+bc*a)*∈ = c*a(d+bc*a)*

Eliminating state q_2, we get-

<u>Question 5:</u> Non- Deterministic Finite Automata **(5+10) Points**

a. Convert the following $\epsilon - NFA$ to DFA

Marks Distribution

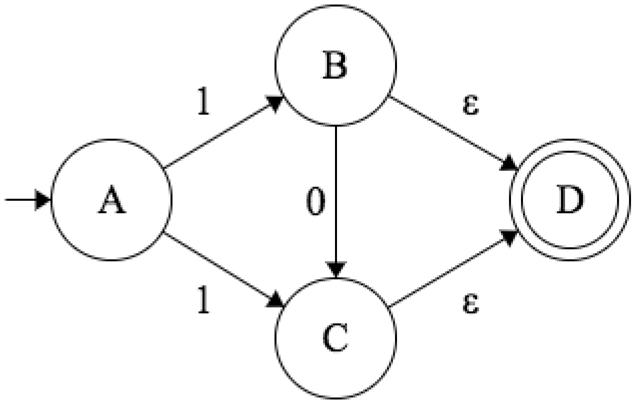
1 if wrong.

3 If no DFA and no initial and final states

4 if No DFA

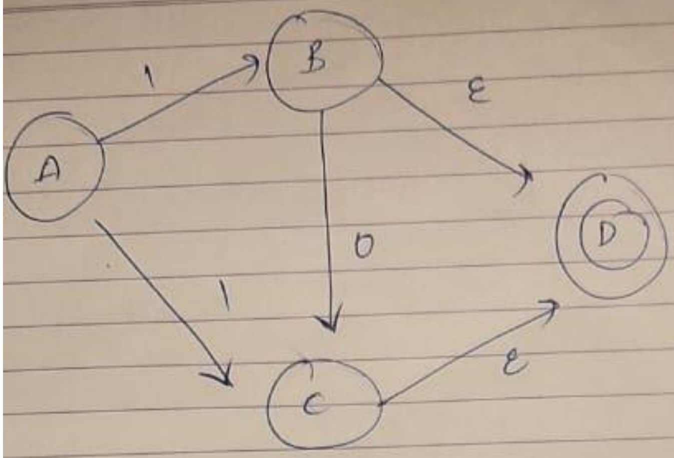
4 if no proper final states and initial state

5 if correct



Date:

Sun Mon Tue Wed Thu Fri Sat

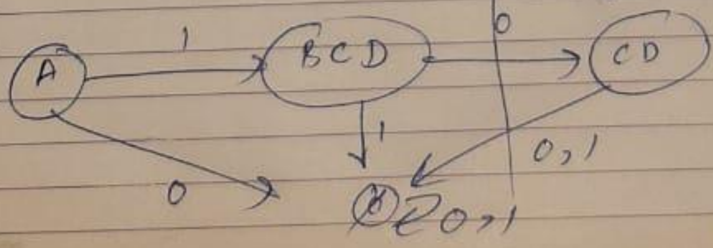


Transition table

states	0	1
A	∅	{B,C,D}
B	{C,D}	∅
C	∅	∅
D	∅	∅

DFA Table

states	0	1
A	∅	{B,C,D}
{B,C,D}	{C,D}	∅
{C,D}	∅	∅
∅	∅	∅



b. Draw the Non-Deterministic Finite Automata based on the following Scenario.

Marks Distribution

2 if states not equal to 5, even if student has drawn NFA

5 states 2

Final state 1

Initial 1

Transitions if complete 6 else 3.

Enemy AI

Finite State Machines allows us to map the flow of actions in a game's computer-controlled players. Let's say we were making an action game where guards patrol an area of the map. We can have a Finite State Machine with the following properties:

States: For our simplistic shooter we can have: Patrol **(P)**, Attack **(A)**, Reload **(R)**, Take Cover **(TC)**, and Deceased **(D)**.

Initial State: As it's a guard, the initial state would be Patrol.

Accepting States: An enemy bot can no longer accept input when it's dead, so our Deceased state will be our accepting one.

Alphabet: For simplicity, we can use string constants to represent a world state: Player approaches **(PA)**, Player runs **(PR)**, Full health **(FH)**, Low health **(LH)**, No health **(NH)**, Full ammo **(FA)**, and Low ammo **(LA)**.

Transitions: As this model is a bit more complex than traffic lights, we can separate the transitions by examining one state at a time:

Patrol

If player approaches go to the Attack state.

If we run out of health, go to the Deceased state.

Attack

If ammo is low, go to the Reload state.

If ammo is low, go to the Diseased.

If health is low, go to the Take Cover state.

If the player escapes (runs), go to the Patrol state.

If we run out of health, go to the Deceased state.

Reload

If ammo is full, go to the Attack state.

If health is low, go to the Take Cover state.

If we run out of health, go to the Deceased state.

Take Cover

If health is full, go to the Attack state.

If health is full, go to the Reload state.

If ammo is low, go to the Reload state.

If we run out of health, go to the Deceased state.