National University of Computer & Emerging Sciences, Karachi

Spring 2020 - CS-Department

Final

9:00 am – 12:30 pm

Course Code: CS301	**Course Name: Theory of Automata**
Instructor Names:	**Shaharbano, M. Shahzad, Bakhtawar**
Student Roll No:	**Section No:**

Solution Paper

Instructions:

- This is an open book/notes exam, read all the questions carefully and don't ask for any clarifications.
- In case of any ambiguity, you may make assumptions. But your assumption should not contradict any statement in the question paper.
- You will attempt this paper **offline**, in your **hand writing**.
- Please solve questions in the order in which they have been posed
- You may use **cam-scanner, MS lens** or any equivalent application to scan and convert your hand-written answer sheets in **a single PDF file**
- The paper should be submitted using **Google Classroom**. You are given 30 minutes for this purpose, which is already included in the exam time mentioned above. Additionally, after submitting, you should email it to your instructor which should be exactly same **pdf** as uploaded earlier.
- All students must use your name with roll# for the pdf file name (e.g. Ahmed-K181245)
- In case of any plagiarism with any of your class fellows found in your solution, you will be punished with **-10 marks (minus ten marks) in that part of the question.**
- Marks of each question are mentioned with each question.
- **WRITE YOUR ID ON TOP OF EVERY PAGE by your hand**. Write also **page # on every page**. **You should also sign on every page.**

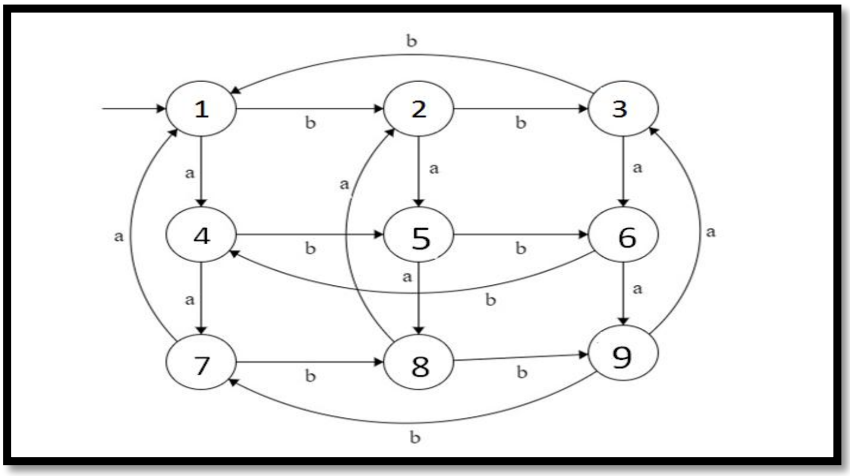
Time: 3.5 Hrs. **Max Marks** : 50 points

Question 1 (Pumping Lemma & DFA Minimization) **(5+5 Points, 20 + 20 mins)**

i. Use pumping lemma to show that whether L ={ $a^i b^{3i}$ | i≥1000 and i≤5000} is non-regular or regular. Show your steps against each of the pumping lemma claims.

Ans: As the given language is regular because of the finite length of strings, therefore no need to use pumping lemma in this case.

ii. In the given DFA mark the final states by using all non zero digits of your four digit roll number.(For Roll # K18-1019, accepting states will be 1 and 9). Now minimize your after deciding the accepting state(s) in the given DFA. Show Proper working and draw the minimized DFA.



Ans:

Let suppose final states are { 1, 9}

		a	b
Partition I	2	5 I	3 I
	3	6 I	1 II
	4	7 I	5 I
	5	8 I	6 I
	6	9 II	4 I
	7	1 II	8 I
	8	2 I	9 II
Partition II	1	4 I	2 I
	9	3 I	7 I

		a	b
Partition I	2	5 I	3 IV
	4	7 III	5 I
	5	8 IV	6 III
Partition III	6	9 II	4 IV
	7	1 II	8 IV
Partition IV	8	2 I	9 II
	3	6 III	1 II
Partition II	1	4 I	2 I
	9	3 IV	7 III

		a	b
Partition I	2	5 VI	3 IV
Partition V	4	7 III	5 VI
Partition VI	5	8 VII	6 III
Partition III	6	9 II	4 V
	7	1 VIII	8 VII
Partition VII	8	2 I	9 II
Partition IV	3	6 III	1 II
Partition VIII	1	4 V	2 I
Partition II	9	3 IV	7 III

		a	b
Partition I	2	5	3
Partition V	4	7	5
Partition VI	5	8	6
Partition III	6	9	4
Partition IX	7	1	8
Partition VII	8	2	9
Partition IV	3	6	1
Partition VIII	1	4	2
Partition II	9	3	7

Hence the given model is already minimized.

Question 2 (Context Free Grammar) **(10 Points, 30 mins)**

The specifications of your target language (MagicL) is as follows:

The alphabet ∑ set contains five symbols, which will be defined as follows:

(1) first letter of your first name

(2) the last letter of your first name

(3) the first letter of your last name

(4) the last letter of your last name

(5) the character $.

For example, if name is "*Ahmed Zubair*" so, in this case, the alphabet for MagicL is **∑={a, d, z, r, $}**.

Clearly the alphabet for your language is likely to be different.

Note: If your name is like "nina nolan" and does not give **4** distinct letters, then choose the alternative letter(s) in consecutive order from your name, so if you were named as in the example you could use **∑= {n, a, o, l, $}** as your alphabet.

Strings of your language must satisfy the following constraints:

The first is that there is exactly one $ in any correct string, and this is always the last symbol in the string. The string before the $, which must be non-empty, we call a **correct expression**, defined as follows:

- A sequence of one or more of the second symbol (2) and/or third symbol (3) in any order, bracketed by the first symbol (1) and fourth symbol (4), OR
- A correct expression bracketed by the first (1) and last letter (4), i.e. for every letter (1) in the string, there must be a unique corresponding matching last letter (4). OR
- A sequence/repetition of correct expressions.

Examples of Correct Strings:

So for my language MagicL, examples of correct strings in the language are (separated by semi colons): **adzzr$; aaadzzrrr$; aazzdzddzdddzrr$; aaadddrdzzrr$; adraddr$; aaazraddzrrr$**.

Examples of Incorrect Strings:

Strings of the alphabet which are NOT in the language include: aadrrar$ (no “d” or “z" inside the last “ar”); aaadrrz$ (missing “r” at the end); aaadrrrrzzddr$ (the fourth “r” and last "r" do not have a corresponding starting “a”).

You are required to:

(i) Create your alphabet set using your full name.

(ii) create a context-free grammar which generates exactly the language described above using your name;

(iii) use your grammar to generate a parse tree of a chosen string in your language which uses all 5 letters of the alphabet;

Note: Your grammar should be small, consisting of not greater than 6 production rules.

Ans:

For suppose name is : Zubair Ahmed

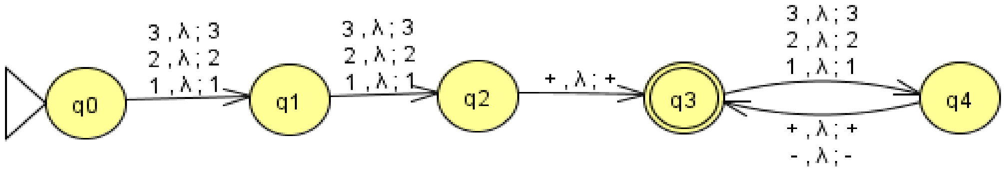
alphabet set will be {z,r,a,d}

CFG
S ->A$
A -> zAd| zBd, | AA
B-> BB|r|a

Question 3 (Pushdown Automata) (10 Points, 15 mins)

Construct a pushdown automata for accepting a postfix notation. Also give one example on your own choice and show all stack updates in accepting or rejecting the input notation. Alphabet ∑ of the language is {1,2,3, +, -}.

Ans:



`A-B+C+A+K-L => AB-C+A+K-L`

AB-
(AB-)C+
((AB-)C+)A+
((AB-)C+)A+
(((AB-)C+)A+)
(((AB-)C+)A+)K+
((((AB-)C+)A+)K+)L-

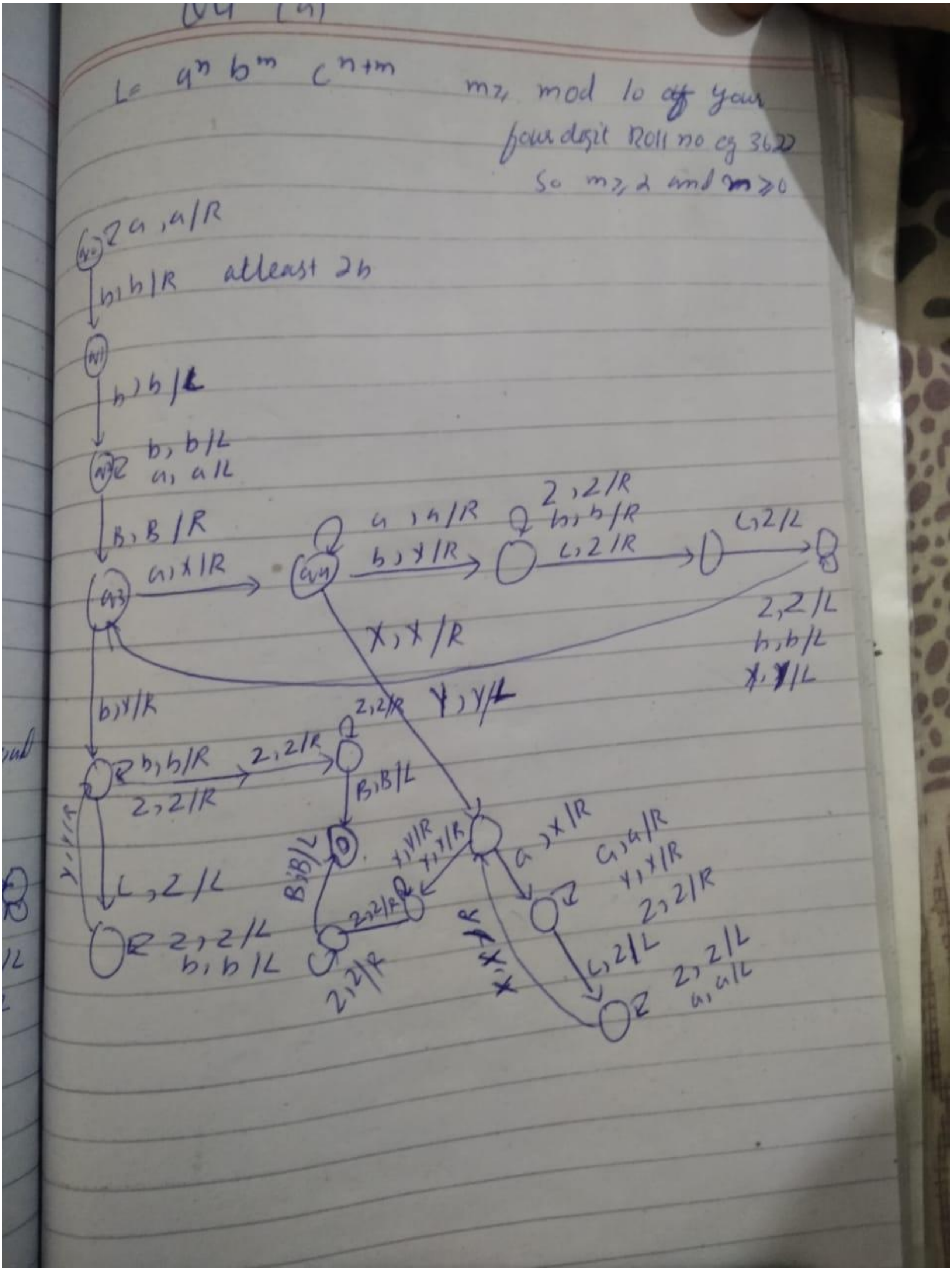
Input String: 2 3 +

Stack updates:

			+
		3	3
	2	2	2
$	$	$	$

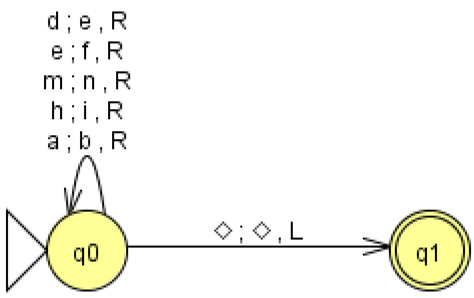
Question 4 (Turing Machines) **(5+5 Points, 20 + 25 mins)**

i. $L_1 = \{a^n b^m c^{n+m} \mid m \geq \text{(mod 10 of your four digit roll number)}, n \geq 0\}$.



ii. Your target TM reads your name and encrypts it with the last digit of your roll no. Encryption function is as follows:
Each letter in the plaintext is 'shifted' a n number of places down the alphabet, where n is the last digit of your roll no. For example, if n=1, then with a shift of 1, A would be replaced by B, B would become C, and so on. If plaintext letter is Z then with a shift of 1, Z would be replaced by A. This is a kind of cyclic shift function.

Ans:
Let suppose roll no. is: K18-1021



Question 5 (Universal Turing Machines) **(5+5 Points, 20 + 25 mins)**

i. Discuss UTM with respect to its motivation, application, capabilities (expressing power) and limitation. Give only one point on each of the above aspects. Each point must not exceeds the two lines description.

Ans:
Motivation:
The primary **motivation** behind the UTM is to directly simulates an arbitrary Turing machine T which has a single ended tape and just two symbols without loss of **generality**.

Application:
- breakthrough that led to the concept of stored programmer computing device.
- an interpreter, which is a program that takes other programs for input and executes them.

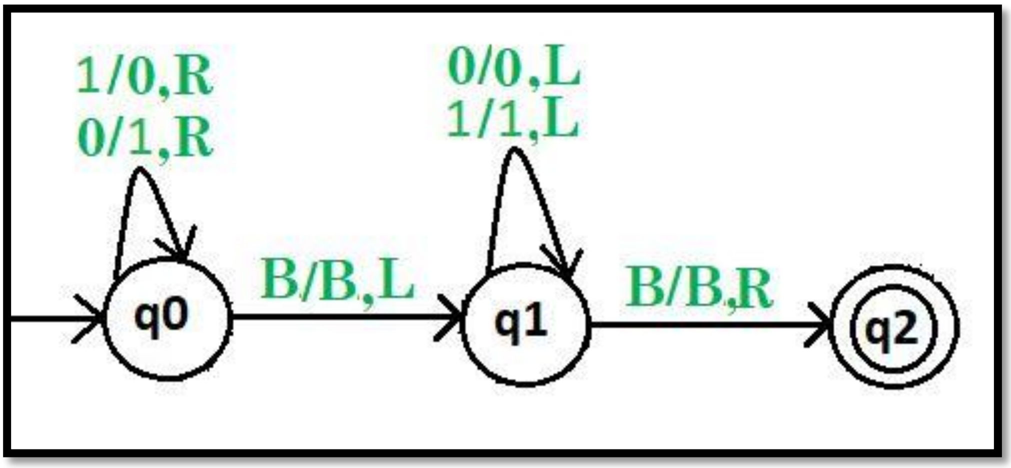
Capabilities:

A **universal Turing machine** can calculate any recursive function, decide any recursive language, and accept any recursively enumerable language.

Limitation:

Parallelism is one the limitations in standard UTM

ii. What is the encoded representation (string) passed to The UTM for this TM.



Q5 (b)

states

q_0 q_1 q_2

1 11 111

alphabets

0 1 B

1 11 111

L R

1 11

transitions

$(q_0, 0) = (q_1, 1 \to R)$ =

(1 0 1 0 1 0 11 0 11)

$(q_0, 1) = (q_0, 0 \to R)$

(1 0 11 0 1 0 1 0 11)

$(q_0, B) = (q_1, B \to L)$

(1 0 111 0 11 0 111 0 1)

$(q_1, 0) = (q_1, 0 \to L)$

(11 0 1 0 11 0 1 0 1)

$(q_1, 1) = (q_1, 1 \to L)$

(11 0 11 0 11 0 11 0 1)

$(q_1, B) = (q_2, B \to R)$

(11 0 111 0 111 0 111 0 11)

1 0 1 0 1 0 11 0 11 00 1 0 11 0 1 0 1 0 11 00

1 0 111 0 11 0 111 0 1 00 11 0 1 0 11 0 1 0 1

00 11 0 11 0 11 0 11 0 1 00 11 0 111 0 111 0 111 0 11

****END OF QUESTIONS****