National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Theory of Automata**	**Course Code:**	**CS-3005**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2023**
Duration:	**180 Minutes**	**Total Marks:**	**60**
Paper Date:	**27-December-2023**	**Weight**	**40 %**
Section:	**ALL**	**Page(s):**	**12**
Exam:	**Final Term**	**Roll No.**	

Instruction/Notes:
1. Answer in the space provided, showing all the steps.
2. You can take rough Sheets but will not be collected.
3. In case of confusion or ambiguity make a reasonable assumption.
4. Attempt all Questions

CLO 1				CLO 2			CLO 3			CLO 4		
a	b	c	**Total**	a	b	**Total**	a	b	**Total**	a	b	**Total**
3	1	8	**12**	2	10	**12**	4	8	**12**	10	14	**24**

CLO 1 [3 + 1 + 8 =12 Marks]

Question 1:

a) If L_1, L_2 and L_3 are Context free languages and $L_4 = L_1 \cap (L_2 \cup L_3)$. What kind of language will be L_4. (RL, CFL or non-CFL) Explain briefly [3 Marks]

Non CFL
$L_2 \cup L_3$= CFL
$L_1 \cap (L_2 \cup L_3)$= Non CFL

b) **True/ False** Context free languages are closed under difference. [1 Marks]

False

c) Tell whether the following Language is context free (CFL) or non- context free (non- CFL). If it is CFL provide PDA else prove it using Pumping Lemma.[2 + 6 = 8 Marks]

$$L = \{a^i b^j c^i d^{(i+j)} : i, j \geq 0\}$$

same approach we discussed during the class.

CLO 2 [2 + 10 =12 Marks]

Question 2:

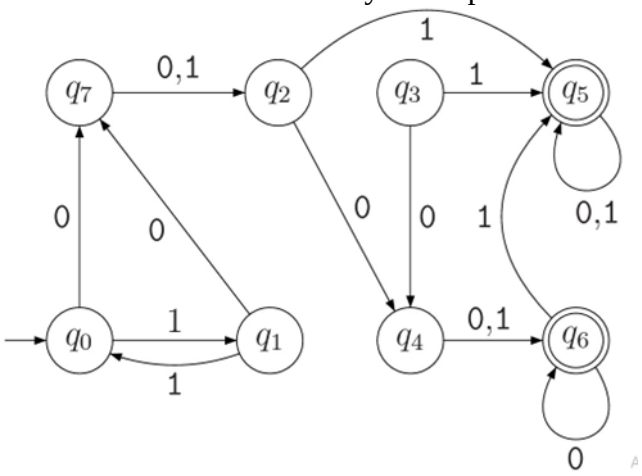
a) **Tick all regular expressions which express** [2 Marks]

L = {w | w ∈ {0,1}* and w has no consecutive 0 and no consecutive 1}

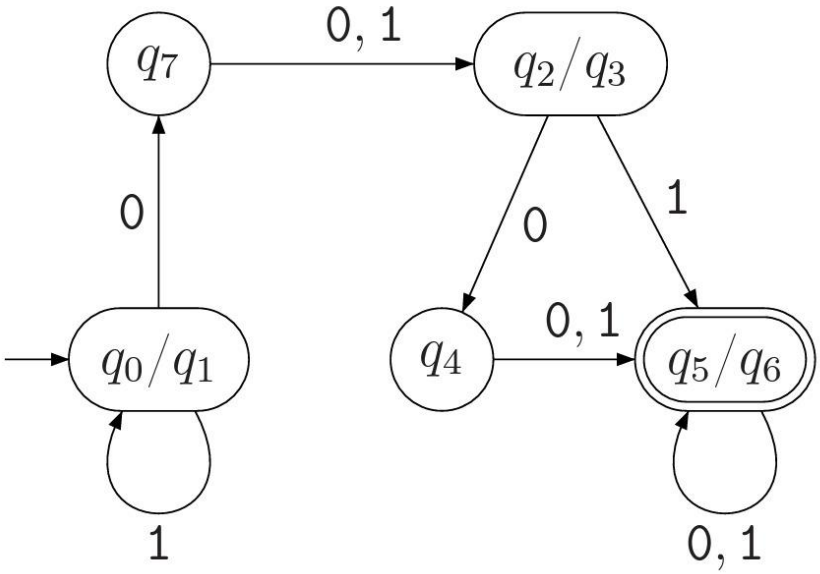
a. (10)*+(01)*
b. (10 + 01)*
c. (0(10)* + 1(01)*)*
d. None of these

b) **Minimization of DFA** [5 Marks]

Find a minimum-state DFA recognizing the same language. Show complete working. Use only the method discussed in your respective classes.



Minimal DFA



CLO 3 [4 + 8 =12 Marks]

Question 3:

a) Tell whether the following grammar is ambiguous. How? Give justification [2+2 = 4 Marks]
S→ 0S1 |0S11 |∧

Ambiguous

Justification: two LMD OR 2 RMD from same string WITH EXAMPLE

b) The Html Table Creator [6 + 1.5 = 7.5 Marks]

Statement:	Sample HTML Code
Consider a context-free grammar that generates Strings in a specific format to create an HTML table. The grammar produces a table with rows%3=0 contains Numbers only, where all other rows can contain a Number or Alphabet. The goal is to design a grammar that generates valid HTML code for such a table structure. Sample Output: 2 | 4 A | 5 1 | D 8 | 9	<table> <tr> <td>2<\td> <td>4<\td> <\tr> <tr> <td>A<\td> <td>5<\td> <\tr> <tr> <td>1<\td> <td>D<\td> <\tr> <tr> <td>8<\td> <td>9<\td> <\tr> <\table>

Design a context-free grammar that generates HTML code for a table with rows%3=0. The multiple of 3 rows should display numbers in each cell, while all the other rows should display alphabet or number. Each row should have at least one cells/columns. Table must contain at least one row. So the minimum number of rows and columns could be a single cell that is one row and one column. Your CFG can have variable number of cells/columns for each row. The generated HTML code should follow the standard syntax and structure of an HTML table. For the context-free grammar, you need to define the production rules that generate the HTML code for the desired table structure. Consider the use of non-terminal symbols for different components of the HTML code, such as the <table>, <tr>, <th>, and <td> tags, as well as the number and alphabet elements. You can use this scenario to design and explore the grammar rules that would generate the desired HTML table structure.

Solution: [Write a CFG for the above scenario also derive the Sample Output [up to 2 rows only] from your CFG using Parse Tree]

CFG:

S → < table > R0 </table > S | < table > R0 </table >
R0 → < tr > C0 </tr > R1 | < tr > C0 </tr >
R1 → < tr > C1 </tr > R2 | < tr > C1 </tr >
R2 → < tr > C1 </tr > R0 | < tr > C1 </tr >
C0 → < td > Num </td > C0 | < td > Num </td >
C1 → < td > Num </td > C1 | < td > Num </td >
C1 → < td > Alpha </td > C1 | < td > Alpha </td >

Num → 0|1|2|3|4|5|6|7|8|9|0
Alpha → a|b| ... |z|A|B ... |Z

Parse Tree:

CLO 4 [10 + 14 =24 Marks]

Question 4:

a) Suppose you have a single tape Turing machine (TM) that is infinite from both ends. The current head (pointer) is at # (in the start of TM). Dry run the TM on next page and give the content of the tape after running it (When TM halts). Also mention the location of the head **[8.5+1.5 = 10 Marks]**

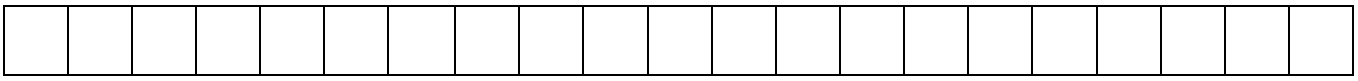
Input String:

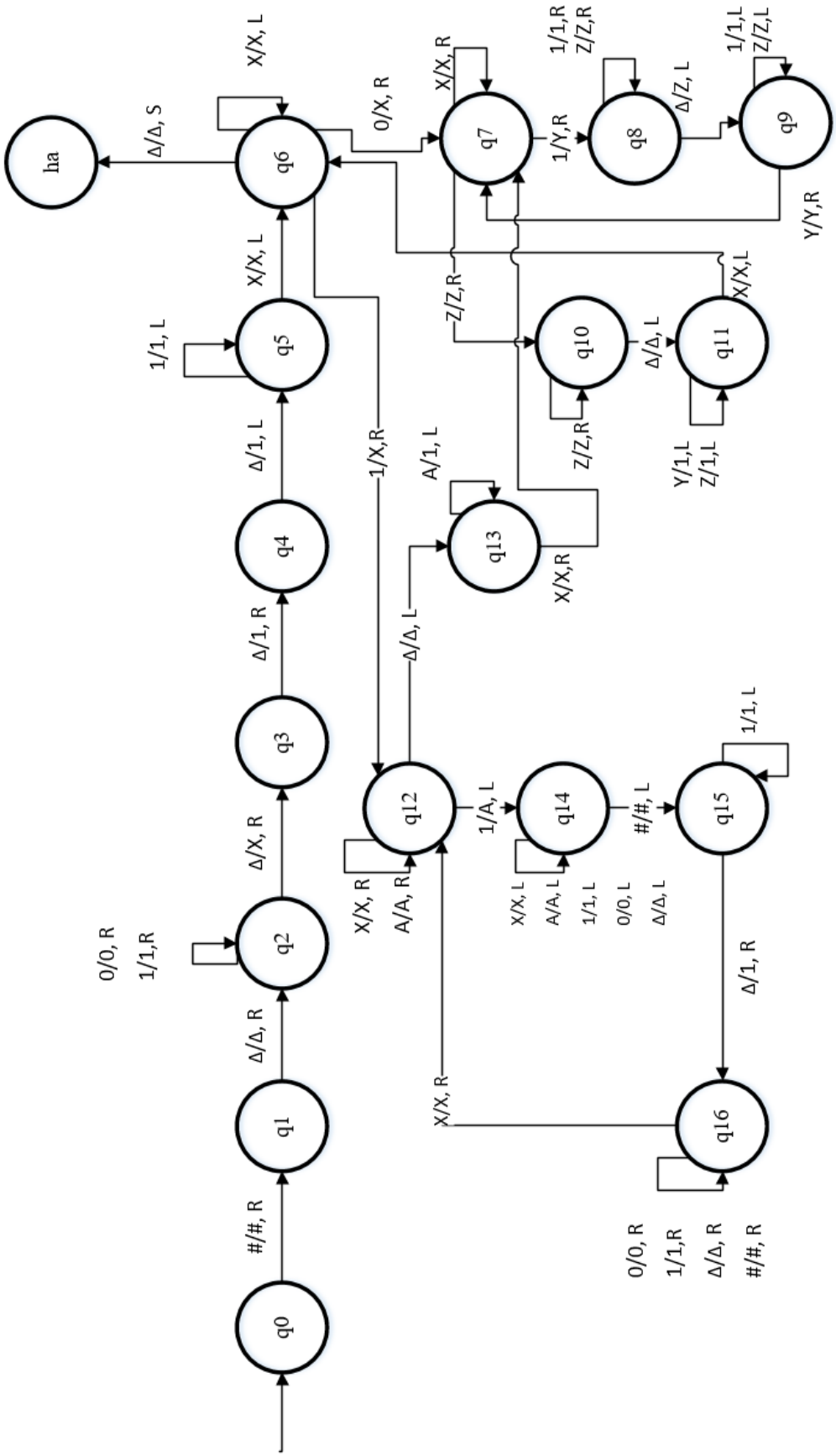
.	.	.	Δ	Δ	#	Δ	1	0	Δ	Δ	Δ	.	.	.

Final Output along with head/pointer Location

ΔΔ1111#ΔXXX11111111ΔΔ

pointer: Δ after hash




q0
q1
q2
q3
q4
q5
q6
ha
q7
q8
q9
q10
q11
q12
q13
q14
q15
q16
#/#, R
Δ/Δ, R
0/0, R
1/1,R
Δ/X, R
Δ/1, R
Δ/1, L
1/1, L
X/X, L
Δ/Δ, S
0/X, R
X/X, R
1/1,R
Z/Z,R
1/Y,R
Δ/Z, L
1/1,L
Z/Z,L
Y/Y,R
Z/Z,R
Δ/Δ, L
X/X,L
Y/1,L
Z/1,L
1/X,R
A/1, L
X/X,R
Z/Z,R
Δ/Δ, L
X/X, R
A/A, R
1/A, L
X/X, L
A/A, L
1/1, L
0/0, L
Δ/Δ, L
#/#, L
1/1, L
Δ/1, R
X/X, R
0/0, R
1/1,R
Δ/Δ, R
#/#, R

b) Design a Multi-Tape Turing Machine that has two inputs X and Y. Both inputs are unary numbers. X is placed on tape 1 while Y is on tape 2. You need to perform X mod Y and store its remainder and quotient in Tape 3 and Tape 4 respectively. Sample input and output is given below. **[14 Marks]**

Note: X and Y are greater than 0.

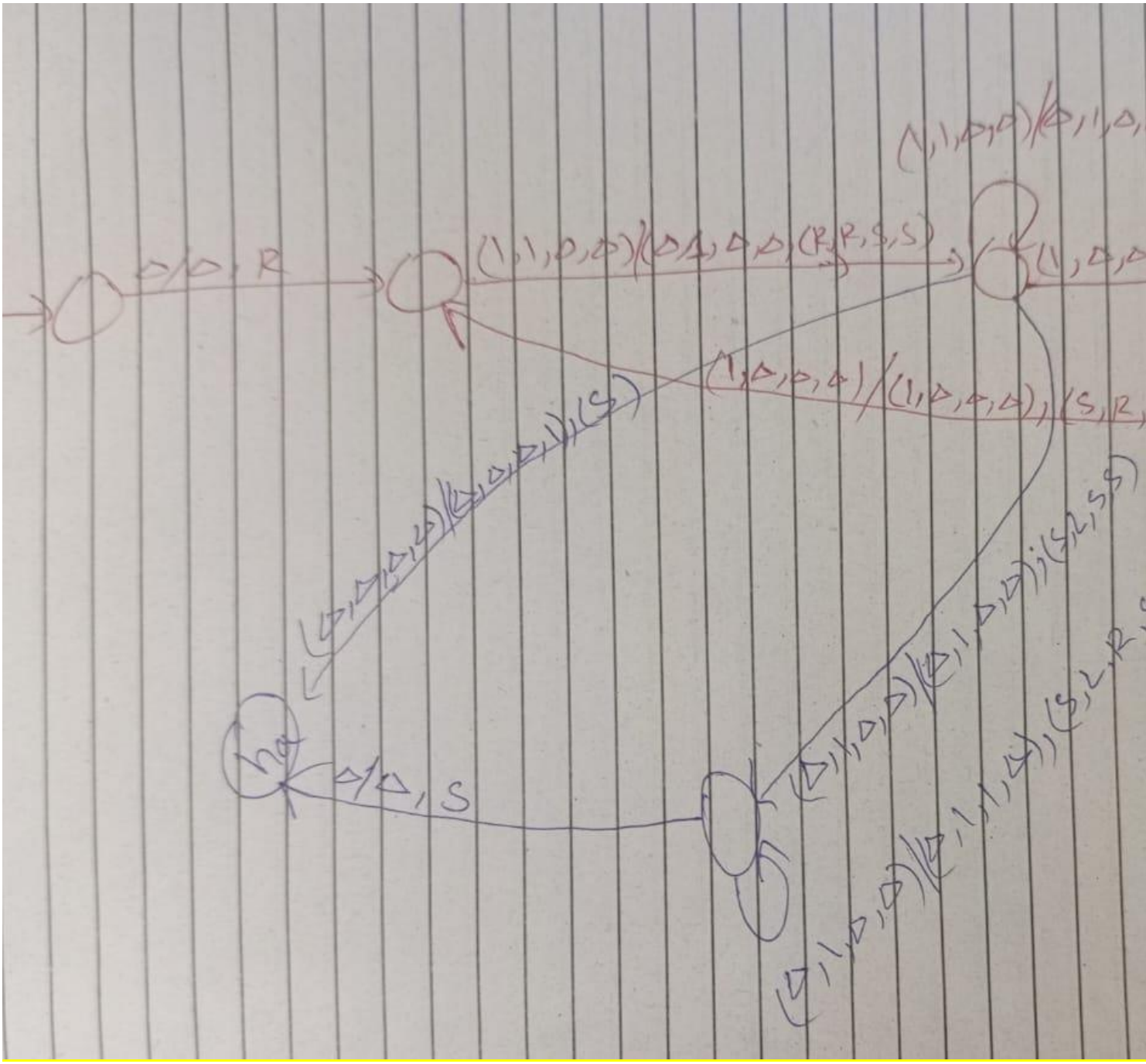
Sample Input:

Tape1	Δ	1	1	1	1	1	Δ
Tape2	Δ	1	1	1	Δ	Δ	Δ
Tape3	Δ	Δ	Δ	Δ	Δ	Δ	Δ
Tape4	Δ	Δ	Δ	Δ	Δ	Δ	Δ

Sample Output:

Tape1	Δ	1	1	1	1	1	Δ
Tape2	Δ	1	1	1	Δ	Δ	Δ
Tape3	Δ	1	1	Δ	Δ	Δ	Δ
Tape4	Δ	1	Δ	Δ	Δ	Δ	Δ

Turing Machine



Rough Work