Z/K/TY

Reg. No: 24MI0285

Final Assessment Test – November 2025

Course: CSI1003 - Formal Languages and Automata Theory

Class NBR(s): 2465/2466/2468 Slot: D2+TD2

Time: Three Hours Max. Marks: 100

- KEEPING MOBILE PHONE/ANY ELECTRONIC GADGETS, EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE
- DON'T WRITE ANYTHING ON THE QUESTION PAPER

COs	CO Statements
CO1	Model, compare and analyse different computational models
CO2	Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.
CO3	Identify limitations of some computational models and possible methods of proving them.
CO4	Explain the abstract concepts mathematically with notations.

BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)

Answer ALL Questions
(10 X 10 = 100 Marks)

1. a) **Prove Using Mathematical Induction** [5] CO4 BL2

Problem:

Let $\Sigma = \{0, 1\}$ be an alphabet. For any string $w \in \Sigma^*$, define the **reversal operation** w^R as the string w written in reverse order.

For example:

- If w = 01101, then w^R = 10110
- If w = ε (empty string), then $w^R = \varepsilon$

Prove by Mathematical Induction that:

For any two strings $x, y \in \Sigma^*$, the following property holds:

$(xy)^R = y^R x^R$

That is, the reversal of the concatenation of two strings equals the concatenation of their reversals in **reverse order.**

b) **Problem:** Language Operations

Let L_1 and L_2 be two languages over alphabet $\Sigma = \{a, b\}$.

Given:

- $L_1 = \{a^n b \mid n \geq 0\} = \{b, ab, aab, aaab, ...\}$
- $L_2 = \{b^n \mid n \geq 1\} = \{b, bb, bbb, ...\}$

Tasks:

i. Find $L_1 \cup L_2$ and $L_1 \circ L_2$. Describe both in set notation. [2]

ii. Find L_1^* (Kleene closure of L_1). Describe the language. [1]

iii. Is concatenation commutative for these languages? i.e., Does $L_1 \circ L_2 = L_2 \circ L_1$? Prove or give a counter-example. [2]

2. a) **Problem:** Convert the following NFA to an equivalent DFA using the subset construction method. [5] CO2 BL3

Given NFA:

- **Alphabet:** $\Sigma = \{0, 1\}$
- **States:** $Q = \{q0, q1, q2, q3\}$
- **Start State:** q0
- **Accept State:** $F = \{q3\}$

NFA Transition Table:

State	Input '0'	Input '1'
q0	{q0, q1}	{q0}
q1	∅	{q2}
q2	∅	{q3}
q3	{q3}	{q3}

Required:

i. Show the DFA transition table with all reachable states
ii. Clearly identify the start state and accept states
iii. Show your working for computing transitions of subset states

b) **Problem:** Convert the following **ε-NFA** to an equivalent **NFA** (without ε-transitions) using the epsilon-closure method. [5]

Given ε-NFA:

- **Alphabet:** $\Sigma = \{a, b\}$
- **States:** $Q = \{q0, q1, q2\}$
- **Start State:** q0
- **Accept States:** $F = \{q2\}$

ε-NFA Transition Table:

State	Input 'a'	Input 'b'	ε (epsilon)
q0	{q1}	∅	{q1}
q1	∅	{q2}	{q2}
q2	{q0}	{q1}	∅

Required:

i. Compute the **ε-closure** for each state
ii. Construct the equivalent **NFA transition table** (without ε-transitions)
iii. Identify the **start state** and **accept states** of the resulting NFA.
iv. Show your complete working.

3. a) Formally define the language acceptance of epsilon NFA. [2] CO2 BL3

b) **Problem: Minimize the following DFA using the** table-filling algorithm **(or equivalence class method).** [8]

Given DFA:

- ***Alphabet:*** $\Sigma = \{0, 1\}$
- ***States:*** $Q = \{q0, q1, q2, q3, q4, q5, q6\}$
- ***Start State:*** q0
- ***Accept States:*** $F = \{q2, q5, q6\}$

DFA Transition Table:

State	Input '0'	Input '1'	Accept?
q0	q1	q3	No
q1	q2	q4	No
q2	q2	q5	Yes
q3	q4	q6	No
q4	q5	q2	No
q5	q5	q2	Yes
q6	q6	q5	Yes

Required:

i. Show the **distinguishability table** or **partition refinement process**
ii. Identify all **equivalent states**
iii. Construct the **minimized DFA transition table**
iv. State the number of states in the minimized DFA
v. Show your complete working step-by-step

4. **Problem:** Consider the Regular Expression: **CO2** **BL3**

$r = (a+b)aba(a+b)$

Construct an **ε-NFA** from the given regular expression using **Thompson's Construction** method, then convert it to a **DFA** using subset construction, and finally obtain the **minimal DFA** using the minimization algorithm. Your solution must include all intermediate steps: ε-NFA with all states and transitions, ε-closure computations, DFA transition table and diagram, state partitioning for minimization, and the final minimal DFA.

Note: Label states clearly (q_0, q_1, etc.) and show all working systematically.

5. a) **Problem:** Consider the language $L = \{a^n b^m \mid n = m^2, n,m \geq 1\}$ over the alphabet $\Sigma = \{a, b\}$. Use the **Pumping Lemma for Regular Languages** to prove that L is **NOT regular.** **[4]** **CO3** **BL2**

b) **Problem:** Consider the following DFA with states Q = {q0, q1, q2}, alphabet Σ = {a, b}, start state q0, and accept state q2: **[6]**

Transition Table:

State	Input 'a'	Input 'b'
q0	q1	q0
q1	q1	q2
q2	q1	q0

Convert the given DFA into a **Regular Expression** using **Arden's Theorem.** Show all equations and steps systematically.

6.a) i. **Scenario:** A software company is developing a parser for a new educational programming language called "EduCode". The language design team proposed the following grammar for conditional statements: [6] CO2 BL3

S → if B then S
S → if B then S else S
S → a
B → b

Where: S = statement, B = boolean condition, a = assignment, b = boolean expression.

Prove that the given grammar is **ambiguous** by demonstrating a statement that has two different leftmost derivations. Show both leftmost derivations clearly and explain the practical problem this creates for the compiler.

Verify: Consider the statement `if b then if b then a else a else a`

ii. **Problem:** Consider the language **L = {ww | w ∈ {a, b}*}** (The language of strings that consist of two identical consecutive substrings). [4]

Examples:

- "aa" ∈ L (w = a)
- "abab" ∈ L (w = ab)
- "abaaba" ∈ L (w = aba)
- "aba" ∉ L

Use the **Pumping Lemma for Context-Free Languages** to prove that L is **NOT context-free.** Your proof must include assumption, string selection, case analysis, and contradiction.

OR

6.b) **Problem:** CO3 BL2

Consider the following Context-Free Grammar G in **Chomsky Normal Form (CNF):**

S → AB | BC
A → BA | a
B → CC | b
C → AB | a

Use the **CYK (Cocke-Younger-Kasami) algorithm** to determine whether the string **w = "baaba"** belongs to the language L(G). Your solution must include a brief explanation of the CYK algorithm approach, the complete CYK parsing table showing all intermediate steps for each cell with proper justification of entries, and a clear conclusion stating whether w ∈ L(G) or w ∉ L(G).

Note: Construct the table systematically showing how each cell [i,j] is computed, where i represents substring length (1 to n) and j represents starting position.

7. **Problem:** Consider the following Context-Free Grammar G: **CO3 BL2**

$S \rightarrow AB$
$A \rightarrow BS \mid 1$
$B \rightarrow SA \mid 0$

Convert the given grammar into **Greibach Normal Form (GNF)**. Your solution must include: GNF definition and requirements, identification and elimination of left recursion (if any), conversion to proper GNF form with all intermediate steps clearly shown, and verification that the final grammar satisfies GNF requirements.

Note: In GNF, every production must be of the form $A \rightarrow a\alpha$ where 'a' is a terminal and α is a string of non-terminals (possibly empty). Show all substitutions and transformations systematically.

8. **Scenario:** Assume a laboratory automation system called ***"ChemSafe"*** validates molecular formulas using a specific notation: **CO1 BL3**

Notation:

- **'A' (Atom):** Represents individual atom units in a molecule. Each atom has a valency of 2, meaning it can form exactly 2 bonds with other atoms.
- **'B' (Bond):** Represents bond connections between atoms. In this simplified notation, bonds are counted explicitly.

Validation Rules:

i. All atoms must be declared first in the formula
ii. All bonds must come after atom declarations (no mixing allowed)
iii. For every atom 'A', there must be exactly 2 bonds 'B' (valency constraint)
iv. Valid formula structure: A...A B...B (atoms followed by bonds)
v. At least one atom must be present

Examples:

- "ABB" → 1 atom with 2 bonds ✓
- "AABBBB" → 2 atoms with 4 bonds ✓
- "AAABBBBBB" → 3 atoms with 6 bonds ✓
- "AABBB" → 2 atoms with only 3 bonds X (violates valency)
- "ABAB" → mixed order X (violates rule 1)

a) Based on the above description:

i. Identify and write the formal language L that represents valid chemical formulas. **[2]**

ii. Design a Pushdown Automaton (PDA) to accept this language. Specify the 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ and provide the complete transition function δ. **[4]**

b) The system receives a formula **w = "AAABBBBBB"** for validation. Trace the complete execution of your PDA by showing all Instantaneous Descriptions in the format **(state, remaining_input, stack_contents)** from initial configuration to final configuration. **[4]**

9.a) A computational system uses a Turing Machine as a **transducer** to perform multiplication using the **COPY subroutine** approach. CO3

Problem:

Design a TM that implements $L = \{0^m\ 1\ 0^n \rightarrow 0^{mn}\}$

Description:

- **Input:** $0^m\ 1\ 0^n$ (m zeros, separator 1, n zeros on the tape)
- **Output:** 0^{mn} (product m × n as zeros)
- **Strategy:** Copy the first segment (0^m) exactly n times using COPY subroutine.

Examples:

- Input: "0010" → Output: "00" (1 × 2 = 2)
- Input: "00100" → Output: "0000" (2 × 2 = 4)
- Input: "001000" → Output: "000000" (2 × 3 = 6)

Design a Turing Machine as a transducer that performs this multiplication. Your answer must include:

i. Formal 7-tuple: M = (Q, Σ, Γ, δ, q_0, B, F) with complete transition function δ. *[8]*

ii. Trace the computation for input **"00100"** (2 × 2 = 4) using Instantaneous Descriptions showing all major steps from initial configuration to final output "0000". *[2]*

OR

9.b) i. **Problem:** Consider the language $L = \{a^n b^{2n} c^n \mid n \geq 1\}$ *[6]* *CO3* *BL3*

Design a Turing Machine that accepts this language. Your solution must include high-level algorithm/strategy explanation, formal 7-tuple M = (Q, Σ, Γ, δ, q_0, B, F), and complete transition function δ showing how TM handles the "2n" constraint for b's.

ii. State the **Halting Problem** of a Turing Machine and prove that it is **undecidable** using proof by contradiction. **[4]**

10. **Problem:** **CO1** ***BL2***

Consider the following instance of **Post's Correspondence Problem (PCP)** over the alphabet Σ = {a, b}:

Index	List A	List B
1	a	ab
2	ab	baa
3	baa	aa

a) Determine whether this PCP instance has a solution. If a solution exists, provide: **[4]**

- The sequence of indices (indices can be repeated).
- Show the string concatenation for both lists proving they match.

b) State whether the Post's Correspondence Problem is decidable or undecidable in general. Justify your answer. [2]

c) **Problem:**

Analyze and classify the following languages in the **Chomsky Hierarchy.** [4]
Complete the table:

Language	Type (0/1/2/3)	Accepting Machine
$L_1 = \{a^n b^n c^n \mid n \geq 1\}$	?	?
$L_2 = \{ww \mid w \in \{a,b\}^*\}$	?	?
$L_3 = \{a^i b^j \mid i = 2j, j \geq 1\}$	?	?
$L_4 = \{a^n \mid n \text{ is prime}\}$	?	?

Instructions:

- Identify the correct Type (0/1/2/3) for each language.
- Specify the minimum machine/automaton required to accept it.

Note: Each blank (?) is worth 0.5 mark.

⇔⇔⇔ Z/K/TY ⇔⇔⇔