Theory Assignment I

Automata Theory Monsoon 2025, IIIT Hyderabad

August 14, 2025

Total Marks: 45 points | Due date: **28th August, 2025**

General Instructions: All symbols have the usual meanings (example: $\mathbb{R}$ is the set of reals, $\mathbb{N}$ the set of natural numbers, and so on). DFA stands for deterministic finite automata. NFA stands for non-deterministic finite automata. a^* is the Kleene Star operation.

---

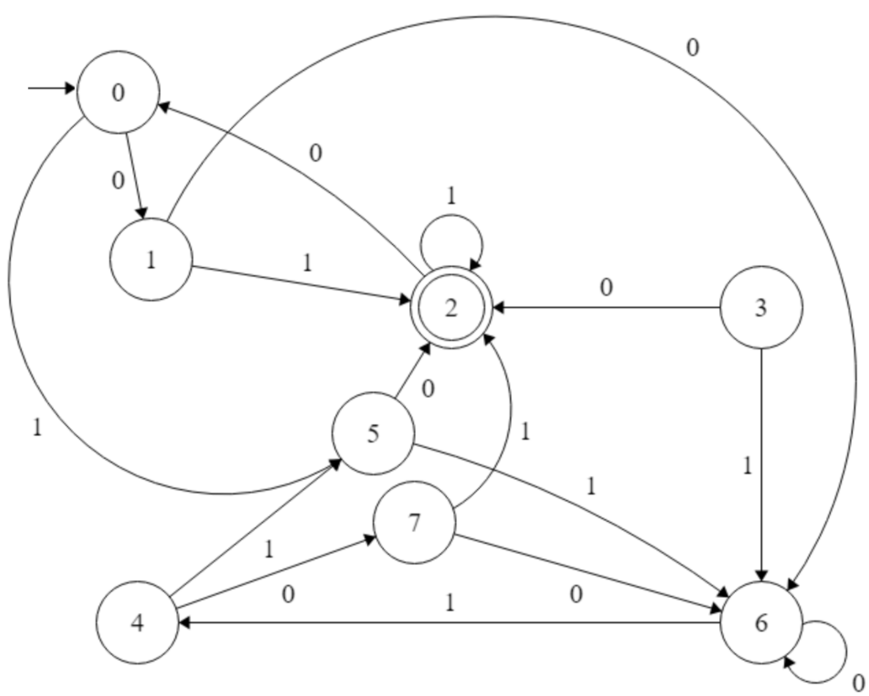
1. [2 points] Prove that if a language L has a finite number of strings, then it is regular. **[CO1, CO3, CO4]**

2. [3 points] Consider a language L over the alphabet $\Sigma = \{0, 1\}$ defined as follows:

$$L = \{w \mid \text{ There exists atleast one length three substring of } w \text{ which is divisible by 3 }\}.$$

Prove that L is a regular language by drawing a DFA for it. **[CO1, CO2]**

3. [3 points] Minimize the DFA given below: [CO1, CO3]



4. [1 + 2 + 2 + 3 points] In this exercise we will prove the *Myhill-Nerode Theorem.*
To prove this statement we will first develop the idea of distinguishable strings. Let L be a language over an alphabet Σ. Two strings $x, y \in \Sigma^*$ are *distinguishable* to L if there is a string z such that $xz \in L$ and $yz \notin L$ or vice versa. If two strings x, y are not distinguishable to L, then we say that they are indistinguishable, i.e. $x \approx_L y$.

   (a) Prove that $\approx_L$ is an equivalence relation over all strings in Σ^*.

   (b) Show that if $x, y \in \Sigma^*$ are distinguishable to L, then for a DFA M that decides L, if M lands on state p while reading x and M lands on state q on reading y, then $p \neq q$.

   (c) A set of strings $\{x_1, ..., x_k\}$ is a distinguishing set of strings for L if all pairs x_i, x_j from this set, are distinguishable to L. If k is the size of this set then, using part (b), show that every DFA for L has at least k states.

   (d) Now prove the main theorem: (i) If Σ^* has infinitely many equivalence classes (with $\approx_L$), then L is not regular. (ii) If the number of equivalence classes is finite (say T many classes), then there exists a DFA M with T many states that can decide L.

   Hint for (d): To prove (i), use part (c) and to prove (ii), give a construction for M. **[CO1, CO2, CO3]**

5. [2 + 2 points] A string y is a *subsequence* of string $x = x_1x_2\cdots x_n \in \Sigma^*$, if there exist indices $i_1 < i_2 < \cdots < i_m$ such that $y = x_{i_1}x_{i_2}\cdots x_{i_m}$. Note that the empty word ϵ is a subsequence of every string. For example, *aaba* is a subsequence of *cadcdacba*, but *abc* is not a subsequence of *cbacba*.

   Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA with language L. Describe a construction for an NFA N' such that:
$$L(N') = \{\, x \mid \text{there is a } y \in L, \text{ such that } x \text{ is a subsequence of } y \,\}.$$

   (A) Explain the idea of your construction.

   (B) Write down your construction formally for the new NFA N'

   [CO1, CO2, CO3]

6. [3 points] Prove that regular languages are closed under Dropout* operation,
$$\text{Dropout*}(A) = \{xz \mid xyz \in A \text{ and } x, y, z \in \Sigma^*\} \tag{1}$$

   [CO1, CO2, CO3, CO4]

7. [3 points] For a symbol a, define $a^+ = \{a, aa, ...\}$. Is the language $L = \{wcw^R | w, c \in \{a, b\}^+\}$ regular? If your answer is yes, write the equivalent regular expression, and if no, prove that L is not regular using the pumping lemma. **[CO1, CO2, CO3, CO4]**

8. [1+1+1 points] Prove that the following grammars are ambiguous by providing a string with two different leftmost/rightmost derivations, or two different parse trees.

   a) $S \to Sa \mid aS \mid a$
   b) $S \to SS \mid a \mid b$
   c) $S \to S + S \mid S \times S \mid x$

   [CO1, CO2, CO3]

9. [2 + 2 + 3 points] A language L is said to be regular if there exists a DFA that decides the language. To show that a language is not regular, we show a contradiction that no DFA can exist for this language. This can be achieved by the pumping lemma.

   Let $\Sigma = \{0, 1\}$. Consider the language $L = \{0^n 1^m | n \leq m \leq 2n\}$. We want to show that no DFA of any arbitrary finite length can exist. We will take you through the proof step-by-step with exercises for you in between.

   To show this, for the sake of contradiction, assume there exists a DFA that decides this language. We then define the pumping length (number of states in the DFA) to be n.

   (i) Find a string s, in the language whose length is greater than the pumping length n. [2 marks]

   Since there exists a string in the language whose length in greater than the pumping length n, this means the DFA needs to have a loop for sure. Let us break the DFA into 3 parts,

   1. $u := q_0, \dots, q_i$ is the part before the loop.
   2. $v := q_i, \dots, q_i$ which is the loop, Note: $|v| > 0$
   3. $w := q_i, \dots, q_n$ is the part after the loop

   We need to show that for any split u, v, w of the string, $w = uvw$ there would exist an $i \in \mathbb{N}$ such that $w' = uv^i w \notin L$.

   (ii) List all possible splits u, v, w of the string s. [2 marks]

   (iii) For each of the possible splits, show $\exists i$ s.t. $w' = uv^i w \notin L$. [3 marks]

   DFAs are memoryless and so the machine does not know how many times it has taken the loop. Therefore, if the DFA accepts $w = uvw$ it will also accept $w' = uv^i w$. But since we know $w' \notin L$ this means the DFA doesn't decide the language, which is a contradiction. From these steps, using the pumping lemma, we have that the language L is not regular. **[CO1, CO2, CO3, CO4]**

10. [2+2 points] For the following languages, prove if the pumping properties are satisfied or not, and also if the languages are regular or not.

    (a) $L = \{(10)^p 1^q |\ p, q \in \mathbb{N}, p \geq q\}$.

    (b) $L = \{a^i b^j c^k |\ i = 0 \text{ or } j = k\}$.

    [CO1, CO2, CO3, CO4]

11. [2+3 points] Pushdown automata:

    (a) Construct a PDA for the language, $L = \{\ a^i b^j c^k | i \neq j \text{ or } j \neq k\ \}$.

    (b) If the constructed a PDA which accepts by final state, convert it into a PDA which accepts by empty stack and vice-versa.

    [CO1, CO2, CO3]