Theory Assignment II

Automata Theory Monsoon 2025, IIIT Hyderabad

September 2, 2025

Total Marks: 35 points

Due date: **15th September, 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. CFL stands for Context-Free Language. a^* is the Kleene Star operation. Take $\mathbb{N}$ to be the set of natural numbers $= \{0, 1, 2, 3, ..\}$ and $\mathbb{Z}$ to be the set of integers.

---

1. [2 points] Show that if L is a CFL over a one-symbol alphabet, then L is regular. **[CO-1, CO-2, CO-3]**

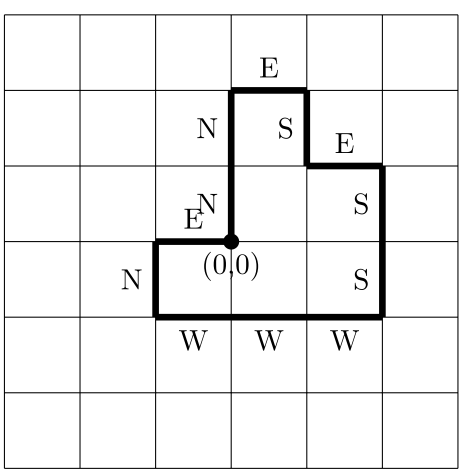
2. [2 points] Prove that language L is not context-free using the pumping lemma.

$$L = \{a^{n!} \mid n \geq 0\}$$

[CO-3, CO-4]

3. [4 points] Prove the stronger form of the pumping lemma for CFL where in the $s = uvxyz$ decomposition the $|vy| > 0$ constraint is replaced by $|v| > 0$ as well as $|y| > 0$. **[CO-2, CO-3, CO-4]**

4. [6 points] Consider the infinite two-dimensional grid, G = $\{$(m, n)$|m, n \in \mathbb{Z}\}$. Every point in G has 4 neighbours, North, South, East, and West, obtained by varying m or n by ± 1.
Starting at the origin (0, 0), a string of commands N, S, E, W generates a path in G. For example, the string NESW generates a path clockwise around a unit square touching the origin. A path is **closed** if it starts at the origin and ends at the origin.



The path taken in the figure is $NNESESSWWWNE$, which is closed

Let $\Sigma = \{N, S, W, E\}$ and $C = \{\omega | \omega \in \Sigma^* \ s.t. \ \omega$ forms a closed path$\}$.

(i) Give a clear description of the language C. [1 point]

(ii) Give two CFLs A and B s.t. $C = A \cap B$. Clearly write down the CFGs for generating A and B. [2 points]

(iii) Is the language C context-free or not? If it is a CFL, construct a PDA that recognizes it. Else using the pumping lemma, prove the language is not context-free [3 points]

[CO-1, CO-2, CO-3, CO-4]

5. [3 points] Design a total Turing machine that accepts the language $\{ 0^{2^i} \mid i \geq 1 \}$. **[CO-1, CO-2, CO-3]**

6. [3 points] Suppose that $\mathcal{C}$ is a proper, non-empty subset of the set of all recursively enumerable languages. Show that, the language $L_{\mathcal{C}} = \{\langle M \rangle \mid M$ is a Turing machine and $L(M) \in \mathcal{C}\}$ is undecidable. **[CO-3, CO-4]**

7. [2 points] Show that S $= \{M \mid L(M)$ contains at least 100 elements$\}$ is recursively enumerable? **[CO-1, CO-2, CO-4]**

8. [3 points] Let R_1, R_2 be two recursive languages, and RE_1, RE_2 be two recursively enumerable languages. Classify the following languages with proper justification:

   - $R_1 \cup R_2$
   - $RE_1 \cup RE_2$
   - $RE_1 \cap R_2$

   [CO-3,CO-4]

9. [4 points] We say that a relation $R \subseteq (\Sigma^*)^k$ is *recursive* if the language $L_R = \{\langle x_1, \ldots, x_k \rangle : (x_1, \ldots, x_k) \in R\}$ is recursive. Define Σ_k, for $k \geq 0$, to be the class of all languages L for which there is a recursive $(k+1)$-ary relation R such that

$$L = \{x : \exists x_1 \forall x_2 \ldots Q_k x_k \, R(x_1, \ldots, x_k, x)\},$$

where Q_k is $\exists$ if k is odd, and $\forall$ if k is even. We define $\Pi_k = \mathrm{co}\,\Sigma_k$, that is, Π_k is the set of all complements of languages in Σ_k. The languages belonging in Σ_i for some i constitute the *arithmetical hierarchy.*

(i) Show that $\Sigma_0 = \Pi_0 = \mathbf{R}$, and $\Sigma_1 = \mathbf{RE}$.

(ii) Show that for all i, $\Sigma_{i+1} \supseteq \Pi_i, \Sigma_i$.

(iii) Show that Σ_2 is the class of all languages that can be accepted (not decided) by Turing machines that are equipped with the following extra power: At any point the machine may enter a special state $q_?$, and the next state will be q_{yes} or q_{no} depending on whether or not the current contents in its string are the encoding of a halting Turing machine. Extend this definition to Σ_i for $i > 2$.

(iv) Show that for all i, $\Sigma_{i+1} \neq \Sigma_i$.

[CO-1, CO-2, CO-3, CO-4]

10. [6 points] We will define a Lilliputian Turing Machine (LTM) as one that has the following description: $\langle Q, \Sigma, \delta, \Gamma, q_{start}, q_{accept}, q_{reject}, k\rangle$. Here $k \in \mathbb{N}$ is some fixed number for that LTM. The LTM has a single tape, which is a one-way infinite tape. Each cell is given an address, starting with index 0. So, the cell's addresses are 0, 1, 2, ... and so on. Initially, the tape head starts off at address 0. The transition function is as follows: $\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \mathbb{Z}$.

Each machine comes with a (well paid) Lilliputian, who helps the machine function.

The working of an LTM for $k = 3$ has been described below. Say the tape head is on address $p = 0$. For a δ transition, $\delta(q_i, \{a_1, a_2, a_3\}) = (q_j, \{b_1, b_2, b_3\}, r)$, the Lilliputian inside the LTM M checks if the current state is q_i and if the symbols at addresses p, $p + 1$, $p + 2$ are equal to a_1, a_2, and a_3 respectively. If these conditions are met, it changes the machine's state to q_j and the symbols at those addresses to b_1, b_2, and b_3. After that, he moves the tape head to the address $p + r$, which is a valid address ($r \in \mathbb{Z}$).

For any general k, if the tape head is at cell p, then in transition, the Lilliputian changes cells p, $p+1$, ... $p + k - 1$. If the tape head moves out of bounds in the one way infinite tape, then it just goes to address 0.

Essentially, in one δ transition, the LTM can modify k cells and then after that the tape head can jump to any other cell instantly (relative to the current position). Your task is to prove that the Lilliputian Turing Machine is as powerful as a standard Turing Machine in terms of the languages it can recognize.

Hint: You need to prove that a normal TM can simulate an LTM, and that an LTM can simulate a normal TM. Your answer can be an informal algorithm that describes how these simulations can be done, but the steps should be laid out clearly. **[CO-1, CO-2, CO-3, CO-4]**