Quiz 1

Automata Theory Monsoon 2024, IIIT Hyderabad

26th August, 2024

Total Points: 15 | Time: 45 mins

<u>General Instructions:</u> FSM stands for finite state machine. DFA stands for deterministic finite automata. NFA stands for non-deterministic finite automata. PDA stands for Push Down Automata. CFG stands for context-free grammar. a^* is the Kleene Star operation.

---

1. [2 points] Let $N_a(w)$ denote the number of occurrences of the letter a in a string w. Construct a grammar G such that the language $L(G)$ is a subset of $\{a,b\}^+$ and $L(G) = \{w \mid N_a(w) = 3k, k \in \{0,1,2,\ldots\}\}$. Note that $L(G)$ does not include the empty string.

   Solution:

$$\begin{aligned} S &\to bS \mid aA \mid b \\ A &\to bA \mid aB \\ B &\to bB \mid aS \mid a \end{aligned}$$

2. [3 points] We define the **remove** operation for languages A and B to be

$$A \textbf{ remove } B = \{w \mid w \in A \text{ and } w \text{ doesn't contain any string in } B \text{ as a substring}\}.$$

   Prove that the class of regular languages is closed under the **remove** operation.

   Solution:
   The idea is to find a regular language that contains strings in B as a substring, and remove from A this language. Therefore, define $L_{\text{substr}} = \Sigma^* B \Sigma^*$. Clearly, L_{substr} is regular because it is the concatenation of regular languages. Since regular languages are closed under complement and intersection, $A \setminus L_{\text{substr}} = A \cap \overline{L_{\text{substr}}}$ is also regular. Thus, A **remove** B is regular because it consists of the strings in A that are not in L_{substr}.

3. [1.5+1.5 points] In an *extended* regular expression, we may use the complementation operator ($\neg$) in addition to the three regular operations ($\cup, \circ, *$).
   For example :
   1(0+1)*0 := set of all binary strings that start with 1 and end with 0
   $\neg$ **(1(0+1)*0)** := set of all binary strings that **DON'T** start with 1 and end with 0

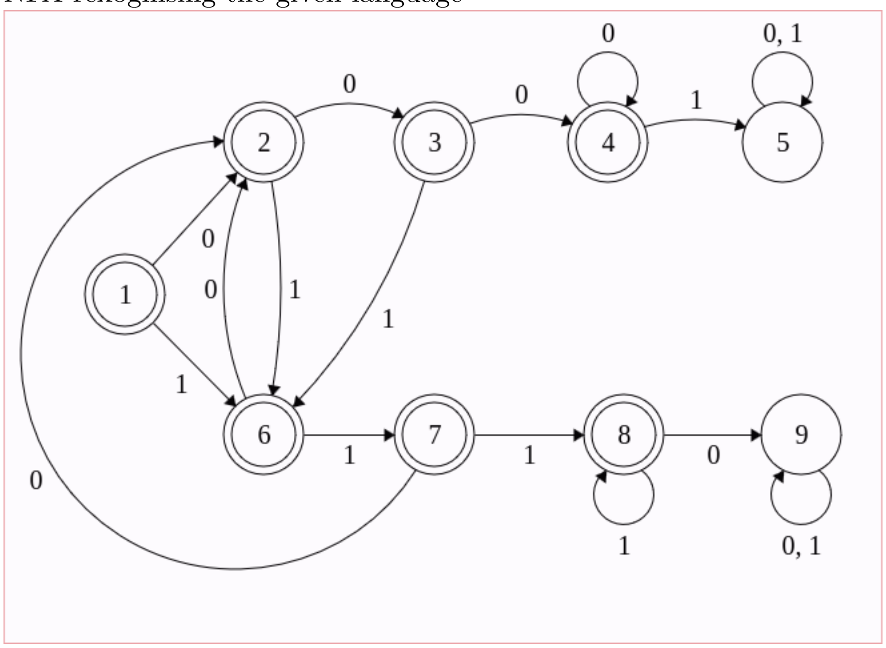
   1. Prove that any language that can be represented by the extended regular expression is a regular language.

2. Construct a NFA that recognizes the following extended regex

$$\neg(\Sigma^*0001\Sigma^*) \cup \neg(\Sigma^*1110\Sigma^*)$$

Hint: You can use the fact that DFA = NFA = Regex = Regular Language

1. Extended regex is just regex + negation. we know a regex is just a representation of regular languages.
Lemma 1: regular languages are closed under complementation
Proof for Lemma 1: Any regular language can be represented as a DFA, and the complement of the language can be found by making all the accepted states of the DFA into normal states and, normal states of the original DFA into the accepting states. Therefore, we can construct a DFA for the complement of the language as well. Therefore, regular languages are closed under complementation.
Since regex is regular, the component over which the negation is performed is also regular. and the negated component is regular.
2. NFA rexognising the given language



4. [3 points] Prove that the language $L = \{a^{2^n} : n \geq 0\}$ is not regular using the pumping lemma.

Solution:
Assume for contradiction that L were regular. Let p be the pumping length, as guaranteed by the pumping lemma. Let $s = a^{2^p}$. Then $s \in L$ and $|s| \geq p$, so, by the pumping lemma, there exist x, y, and z such that $s = xyz$ and $y \neq \epsilon$ and $xy^iz \in L$ for all $i \geq 0$. In particular, for some $\alpha \geq 1$ (namely, $\alpha = |y|$), we have that $a^{2^p+i\alpha} \in L$ for all $i \geq 0$, which means that $2^p + i\alpha$ is always a power of two, for any $i \geq 0$.

Thus (looking at $i = 1$) we have that $2^p + \alpha$ is a power of two, and (looking at $i = 2$) we have that $2^p + 2\alpha$ is a bigger power of two, so it must be at least twice $2^p + \alpha$; that is, $2^p + 2\alpha \geq 2(2^p + \alpha)$, which means that $2^p + 2\alpha \geq 2^p + 2^p + 2\alpha$, so $0 \geq 2^p$, which is impossible.

5. [4 points] Let $\Sigma = \{0, 1\}$. Consider the language $L = \{0^n1^m | n \leq m \leq 2n\}$. Is L regular or context-free? If you think it is regular, write the regular expression for it and draw the corresponding

DFA/NFA. If you think it is context-free, write the grammar rules and draw the corresponding PDA.

Solution:
The language is not regular but is context free and requires a stack to keep in track of the number of 1s. Thus, we shall construct a PDA for this language.

The grammar for this language would be,

$$S \rightarrow 0S1|0S11|\epsilon \tag{1}$$

The corresponding PDA of this language would be,