Theory of Computation '24 Midsem
Time : 120 minutes

Instructions : All solutions have to fit in the 12-page answer sheet provided to you. *No supplementary sheet would be provided.* Feel free to use as many sheets for rough work as you need. This is a closed book/notes/internet exam.

Problem 1. (10 points) We say a string x is a proper prefix of a string y, if there exists a non-empty string z such that $xz = y$. For a language A, we define the following language

$$NOPFX(A) = \{w \in A \mid w \text{ is not a proper prefix of any string in A }\}$$

Show that if A is regular, then so is $NOPFX(A)$. Start with a DFA for A and prove the rest. A simple clean description of the DFA in English is enough. You do not need to prove correctness of the construction (but of course the construction has to be correct !).

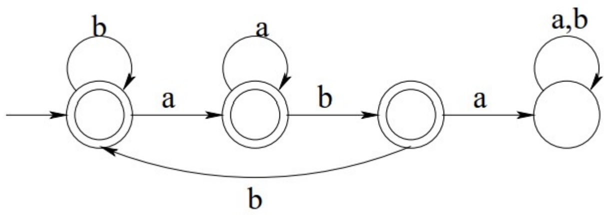
Solution. Let M denote the DFA for A. $NOPFX(A)$ contains strings $w \in A$ (i.e. strings that reach some final state in M), but with the constraint that there must not exist any string x, which can be concatenated with w to again reach a final state in M.
How can we ensure this? We can modify the set F of final states of M. We want $F' \subseteq F$ such that for every state in F', there does not exist a path to any other state in F. Algorithmically, you can check whether some state $f_0 \in F$ has a directed path to some $f_1 \in F$ by running a BFS or DFS (note that f_0 may be equal to f_1, and hence, we also avoid self-loops). Using such a subset F' as the set of final states results in the required DFA for $NOPFX(A)$.

Rubric: +5 for somewhat incorrect but relevant constructions. +10 for correct construction.

Problem 2. (4+6 points) Prove that the following languages are regular, either by exhibiting a regular expression representing the language, or a DFA/NFA that recognizes the language:

1. The set of all strings in $\{a, b\}^\star$ that do not contain aba as a substring

   Solution.

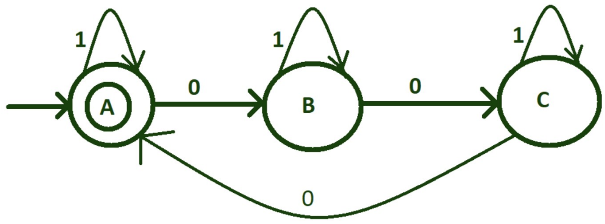
   

   You can also represent the language with the regular expression $b^\star \cup b^\star a^+ \cup (b^\star a^+ bb)^\star$.

   Rubric: +2 for drawing all states correctly and +2 for showing all the transitions correctly.

2. The set of all strings over $\{0, 1\}^\star$ such that the number of 0's is divisible by 3.

   Solution.



Alternatively, you could also use the regular expression $1^\star \cup (1^\star 01^\star 01^\star 0)^\star$.

Rubric: +4 for handling the cases where the number of 0's is positive and +2 for the case where there are no 0's.

Problem 3. (10 points) Define C to be all strings consisting of some positive number of 0's, followed by some string twice, followed again by some positive number of 0. For example 1100 is not in C, since it does not start with at least one 0. However 0001011010000000 is in C since it is three 0's, followed by 101 twice, followed by seven 0's. Prove using the Myhill-Nerode theorem that C is not regular.

Solution. To prove that C is non-regular, we show that an infinite set S is pairwise distinguishable by C. Let us use the set

$$S = \{01^k 0 \mid k \geq 1\}$$

We can distinguish any pair of strings $x, y \in S$, such that $x = 01^p 0$ and $y = 01^q 0$, $p < q$, by the extension $z = 1^p 00$, since

$$xz = 01^p 01^p 00 = 0w^2 0 \in L \quad \text{where } w = 1^p 0$$
$$yz = 01^q 01^p 00 \notin L \quad \text{since } q \neq p$$

where $yz \notin L$ follows from $p \neq q$ because the string $01^q 01^p 00$ cannot be split into the form $0^+ w^2 0^+$ for any choice of substring w. Why? We only have two options to select the trailing 0's, i.e. the string between the 0's can be either $1^q 01^p 0$ or $1^q 01^p$, neither of which can be split as some w^2. Hence, C is not regular. □

Note that there may be simpler/different infinite sets S, which are pairwise distinguishable by C.

Rubric: +4 for choosing *a correct* S, +3 for choosing *an appropriate extension*, and +3 for showing pairwise distinguishability.

Problem 4. (10 points) Let A be the set of strings over $\{0, 1\}$ that can be written in the form $1^k 0y$ where y contains at least k 1s, for some $k \geq 1$. Show that A is not a regular language. (Any correct method of proof will be rewarded)

Solution. We first prove this using the Devil's game format of the Pumping Lemma. Let p be the pumping length given by the Devil. Consider the string $s = 1^p 0^p 1^p \in A$. The Devil gives a split into 3 pieces $s = abc$, where $|ab| \leq p$. Hence, b consists entirely of 1's - say $i \geq 1$ many. But then, $ab^2 c = 1^{p+i} 0^p 1^p \notin B$ and hence you win. □

An alternate solution is to use Myhill-Nerode theorem to prove non-regularity. The following infinite set S is pairwise distinguishable by A

$$S = \{1^k 0 \mid k \geq 1\}$$

For any $x, y \in S$, let $x = 1^p0$ and $y = 1^q0$ such that $p < q$. Then the string $z = 1^p$ is a distinguishing extension, since

$$\begin{aligned} xz &= 1^p01^p \in A \quad \text{since } xz = 1^p0z \text{ and } z \text{ contains at least } p \text{ 1s} \\ yz &= 1^q01^p \notin A \quad \text{since } p < q \end{aligned}$$

This proves A is non-regular. □

Note: you may also choose to prove that A is non-regular by contradiction and closure properties - assume A to be regular and show that applying regular operations on A and some other language results in a non-regular language.

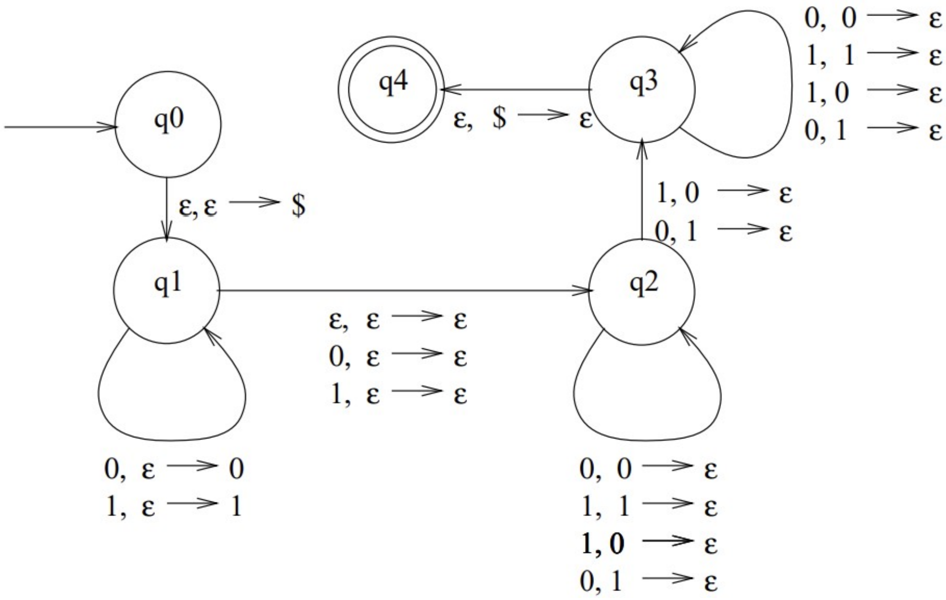
Rubric:

1. **Pumping Lemma: +5 for choosing *a correct* string and +5 for pumping.**
2. **Myhill-Nerode Theorem: +4 for choosing *a correct* S, +3 for choosing *an appropriate extension*, and +3 for showing pairwise distinguishability.**

Problem 5. (10 points) Give a state-diagram-based description of a PDA that accepts the language

$$L = \{x \in \{0,1\}^\star \mid x \text{ is not a palindrome}\}$$

Solution.



Rubric: +5 for the correct states and +5 for correct transitions.

Problem 6. (10 points) Give a simple description in English of the language generated by the following grammar with a short justification, then use that description to give a CFG for the complement of that language.

$$\begin{aligned} S &\to aSb \mid bY \mid Ya \\ Y &\to bY \mid aY \mid \varepsilon \end{aligned}$$

Solution. The grammar generates the complement of $\{a^n b^n \mid n \geq 0\}$. Hence the grammar for the complement of the given language is simply

$$S \to aSb \mid \varepsilon$$

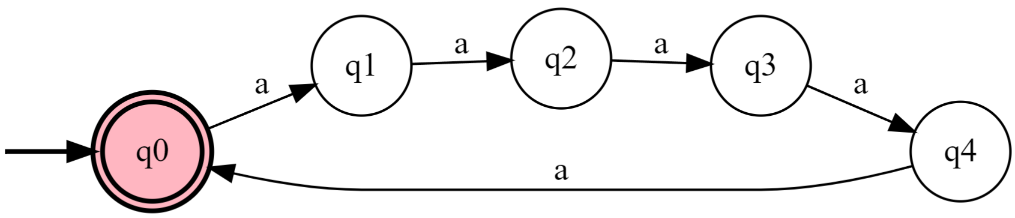
Rubric: +3 for correctly identifying the language and +3 for justification. +4 for second part (only if first part is correct).

Problem 7. (10 points) Design a DFA for the following language with 5 states.

$$L = \{x \in \{a\}^\star \mid |x| \equiv 0 \bmod 5\}$$

Prove the optimality of your construction. In other words, show that any DFA for this language will have at least 5 states.

Solution. We can create the following DFA for L, whose states *remember* the length of the string read so far, modulo 5. In the DFA given below, state q_i represents $|x| \equiv i \bmod 5$.



We can use Myhill-Nerode theorem to prove the optimality of our construction. Use the set

$$S = \{a, aa, aaa, aaaa, aaaaa\} = \{a^n \mid n = 1, 2, \ldots, 5\}$$

Let $x = a^i, y = a^j, 1 \leq i < j \leq 5$. For every such pair (x, y), the string $z = a^{5-i}$ is a distinguishing extension, since

$$\begin{aligned} xz &= a^i a^{5-i} = a^5 \in L \\ yz &= a^j a^{5-i} = a^{5+j-i} \notin L \quad \text{since } 1 \leq j - i \leq 4 \end{aligned}$$

So, S is pairwise distinguishable by L. □

A generic template for a valid distinguishable set can be $S = \{a^i, a^{i+1}, a^{i+2}, a^{i+3}, a^{i+4}\}$ for some fixed non negative i (where $a^0 = \varepsilon$). The distinguishing extension may differ in such cases.

Alternatively, if you have a lot of time, you can apply the DFA minimization algorithm to your DFA, and show that there are exactly 5 distinguishable states. The technique will be similar to the example covered in Lecture 5 where we minimized the DFA for $L = \{x \in \{a\}^\star \mid |x| \equiv 1 \bmod 3\}$. However, in this case, only one minimization step will show you that you cannot merge any states (assuming you actually started with the correct DFA of 5 states).

Another alternate solution can be constructed using the Pigeonhole Principle, where the strings are pigeons and states are holes. The argument is straightforward. By way of contradiction, assume that some DFA for L with (say) 4 states. Any DFA for L must contain a cycle or a self-loop since L is infinite. Since L is constructed using only one alphabet, this means that there can be at most 1 self-loop, which, if constructed, must be at the end of a *linked-list*-like arrangement of the remaining states. Instead of a self-loop, if the DFA contains a cycle, then it must contain at most 4 states.

Due to the cyclic property, you can produce 5 strings such that there will be (at least) 2 strings which end up in the same state q. Simply, you can construct these two strings x and y such that $x \in L$ and $y \notin L$. Then, $q \in F$ and $q \notin F$, which is a contradiction. Otherwise, you may also claim and prove that there exists an extension $z \in L$ that can lead xz and yz both to an accept state, which is also a contradiction ($s_1, s_2 \in L \implies s_1 \cdot s_2 \in L$ by the cyclic nature of the modulus operator).

The same holds for any DFA with 3 states or fewer. $\square$

Rubric: +5 for the correct DFA and +5 for proving optimality - here you get +2.5 for the correct distinguishable set and +2.5 for proving this is pairwise distinguishable. Other style of proofs are up to subjective judgement of grader.