Theory of Computation ’23 Quiz 2
Time : 40 minutes

Name : **Roll :** **Marks :**

Problem 1. Recall the statement of the Pumping Lemma. It says that, for every regular language, *there exists* a number p such that a certain condition is satisfied. For each of the regular languages below, determine the *minimum value* of p for which the lemma would hold. We expect the following from your reasoning. Firstly, you need to determine the correct minimal value $p^\star$. Then you should be able to give an example where a value less than $p^\star$ does not suffice for the lemma to hold.

a) (4 points) $0001^\star$

Solution. The minimum length required here is **4**. A justification of this is that any string of length 4 which is in this language will end with a 1. Hence you can always choose a split xyz where $x = 000$, $y = 1$ and z is the remaining string. It is easy to see that for *any* $i \geq 0$, xy^iz is in the language. (You do not need this justification for credit. This is just for explanation.)

A pumping length of 3 does not work for the lemma. Consider the string 000 which is clearly in the language. But no matter what split you choose, you cannot pump and remain in the language.

b) (4 points) $1^\star01^\star$ The minimum length required here is **2**. A justification of this is that any string of length 2 which is in this language will either begin or end with a 1. Suppose it begins with a 1. Hence you can always choose a split xyz where $x = \varepsilon$ $y = 1$ and z is the remaining. It is easy to see that for *any* $i \geq 0$, xy^iz is in the language. (You do not need this justification for credit. This is just for explanation.)

A pumping length of 1 does not work for the lemma. Consider the string 0 which is clearly in the language. But no matter what split you choose, you cannot pump and remain in the language.

Problem 2. Use Pumping Lemma to prove that the following languages are not regular. Use precise steps as shown in the lectures as much as possible.

a) (4 points) $\{w \in \{0,1\}^\star | w \text{ is not a palindrome } \}$

You can either do this directly or use closure of complement. I will do the latter. We will show that the complement of the language, $L' = \{w \in \{0,1\}^\star | w \text{ is a palindrome } \}$ is non-regular. We will use the Demon game strategy (this is not necessary, any valid use of Pumping Lemma is fine. But, in exam, you will need to stick to the following template for these proofs.)

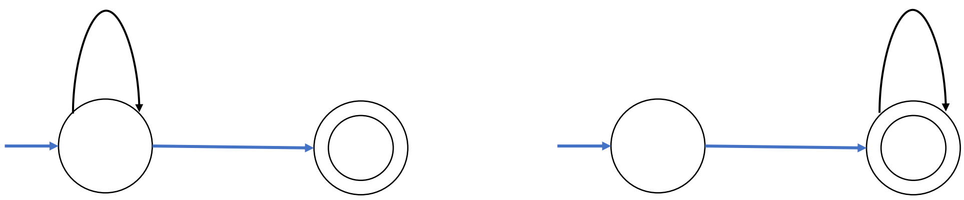
(a) Demon picks p. We choose the string 0^p10^p which is clearly in L'.

(b) Now Demon chooses the split xyz which satisfies the conditions $|xy| \leq p$ and $y \neq \varepsilon$. Hence, no matter what split the Demon chooses, y will be an all-0 string.

(c) Choose $i = 2$. Clearly $0^{2p}10^p$ is not a palindrome and hence not in L'

b) (4 points) $\{wtw | w, t \in \{0,1\}\{0,1\}^\star\}$ Actually the same string as in the above example works for this one as well - 0^p10^p (here $w = 0^p, t = 1$). Again, no matter what split the Demon chooses, y will be an all 0 string. Hence, for $i = 0$ (pumping down), the string will be $0^{p'}10^p$, where $p' < p$ and hence this string is not in the language.

Problem 3. (4 points) Show that for the following regular language $\{w \in \{a\}^\star | w \text{ contains at least one } a\}$, there are two minimal NFAs that are not *isomorphic* (which means that you cannot simply 'relabel' the states of one to obtain the other).



Solution.

The given language can be generated by two regular expressions - $aa^\star$ or $a^\star a$. This gives rise to two minimal NFAs as shown which are not isomorphic to each other. (All arrow symbols are a).
Remark: Note that this example will not work for DFAs. In fact, as we showed in the class that the minimization algorithm always finds the equivalence classes and collapsing these classes gives rise to a unique DFA. This problem shows that such property does not hold for NFAs.