Theory of Computation '23 Midsem
Time : 60 minutes, Marks : 35

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

Beware: Solutions are to be written in the space provided. NO EXTRA SHEET. Meaningless rambles fetch negative credits. $\#_t(w)$ the number of occurrences of a substring t in string w

Problem 1. (10 points) The n-th Fibonacci number is defined as $F_1 = 1, F_2 = 1$, and for all $n \geq 3, F_n = F_{n-1} + F_{n-2}$. Let $\Sigma = \{a\}$. Prove that $L_2 = \{a^m \mid m = F_n\}$ is not regular.

Solution. Suppose that L_2 were regular. According to the pumping lemma, there exists an integer $p \geq 1$ such that every word $w \in B$ of length at least p has a decomposition $w = xyz$, with $|xy| \leq p$ and $y \neq \epsilon$, such that $xy^tz \in B$ for all $t \geq 0$. (+2 for stating that P.L. or pigeon hole is useful)

Now Demon gives you the p (alternately, given any p. You CANNOT CHOOSE p). (+2 for stating this)

Pick some Fibonacci number F_n such that (1) $F_n \geq p$ and (2) $F_{n+1} - F_n > p$. (you can always find such an n given any p). Select the string $s = a^{F_n} \in L_2$ (+2 for choosing a string dependent on p. If you choose any arbitrary string say of constant length etc., you do not get credit)

Now any decomposition of $s = xyz$ has to satisfy $|xy| \leq p$ and $y \neq \epsilon$. Hence $xy^2z = a^m$, where $F_n < m \leq F_n + p < F_{n+1}$ and so m is not a Fibonacci number. (+2 for stating the decomposition and +2 for the final reasoning)

(Any alternate argument/usage of the properties of Fib numbers would be credited for this proof. But the overall proof structure has to be as above. One can also choose to directly use pigeonhole principle instead of explicitly using Pumping Lemma).

Problem 2. (4 points) Prove that $L_2 = \{w \in \{0,1\}^* \mid \#_0(w) \cdot \#_1(w) \text{ is even}\}$ is a regular language.

Solution. This is very straightforward. The marking here is binary. If you have the correct DFA/regular-expression, you get +4, else 0.

Problem 3. (7 points) Let PAREN be the set of all possible balanced parenthesizations. Prove or disprove that PAREN is a regular language.

Solution. Assume PAREN is regular. Hence Pumping Lemma is true. (+1 for stating that Pumping Lemma etc can be used)

Let the Demon give you the integer $p > 0$ (you can also say this as, given a p. You CANNOT CHOOSE a p on your own). (+1 for stating that you are given the p)

Now you select the string $s = (^p)^p$ which is clearly a valid parenthesization and hence belong to the language. (+2 for choosing the string)

No matter what split xyz Demon chooses, the conditions $|xy| \leq p$ and $y \neq \varepsilon$ ensures that y contains one or more (. (+2 for stating the property of any arbitrary decomposition)

Hence, for any $i \geq 0$, xy^iz cannot be a valid parenthesization. (+1 for the closing argument).

Problem 4. (7 points) Show that a single tape Turing Machine can be simulated using a finite automaton and two stacks (without a tape).

Solution. One can imagine the tape to be split in to two parts at the exact position where the head is. We maintain the following invariant The portion of the tape to the left of the head is stored in one stack, say S_Lm with the top-most element being the symbol directly to the left of the head. The portion of the tape to the right is stored in another stack, say S_R with the topmost element being the symbol to the current symbol the head is pointing to. Initially, S_L is empty and S_R contains the input assumed to be in the desired order. (If not, this can be achieved by reversing the entire input using S_L).

A left to right movement of the head is simulated by popping S_R. Depending on whether the TM overwrites this symbol, we push one symbol to S_L. The right to left movement is analogous.

Rubric. +1 for realizing that the tape can be split in to two parts. +2 for properly describing what the stacks are maintaining. +4 for describing left to right and right to left.

Problem 5. (7 points) Prove that $L = \{w \in \{a,b\}^* \mid \#_{ab}(w) = \#_{ba}(w) + 1\}$ is regular. You can do this in **either** of the two ways :

a) Draw a DFA/NFA. You would need to justify correctness (formally or informally). Informal justification means a precise description of what each state is supposed to 'remember' or what it 'signifies' and why each transition does the right thing.

b) Find a simple regular expression for the language. In this case, you would need to formally argue using induction why the language generated by the regular expression is exactly L. No other justification is acceptable. For whatever solution you give, you get 4 points for the correct DFA/NFA or regular expression and 3 points for the justification.

Solution. $L = L(a(a+b)^*b)$. Let us prove the claim by induction. Firstly notice the shortest string in the language must have length at least 2 since the shortest string in L should have at least one ab. This is because the lowest $\#_{ba}(w)$ can be is 0. Notice also that any string $w \in L$ has to begin with an a and end with a b (Why? Try to convice yourself with a short case analysis of strings of different forms like $w = bx$ or $w = axa$, where $x \in \Sigma^*$). Let us prove by induction on length of w that $L = a(a+b)^*b$. For the base case $|w| = 2$, ab is the only string in the language. Let us assume as the induction hypothesis that for all strings of the form $w = axb \in L$ where $x \in \{a,b\}^*$ we have $\#_{ab}(w) = \#_{ba}(w) + 1$. For the induction step there are two cases $z = wa$, in this case, $\#_{ab}(z) = \#_{ab}(w) + 0 = \#_{ba}(w) + 1 = \#_{ba}(z)$ and hence $z \notin L$ as required. For the case when $z = wb$, $\#_{ab}(z) = \#_{ab}(w) + 0$ (since w should have ended in a b) and therefore $\#_{ab}(z) = \#_{ba}(z) + 1$ which just follows from the induction hypothesis. Alternately, one can draw a DFA for L to show it is regular as well, and simplifying those might lead to many different regular expressions (but the one we have given here is probably the simplest).

Rubric. Any solution with the above regular expression or an equivalent regular expression / DFA / NFA with some explanation gets full credit. In a few cases I have awarded partial score if there is a small mistake. That judgement is totally subjective.