Theory of Computation Homework 1

313782 Gregory Richard
324944 Damian Kopp
350510 Kim Yujun
329975 Khalil El Ajeri
Yiran Li

March 2022

Problem 1

Claim: If the input string x with length l is the empty word, only contains 0s, or ends with "00", M finishes in $q1$. If x ends with "1" and not "101", M finishes in $q2$. If x ends with "10", M finishes in $q3$. If x ends with "101", M finishes in $q4$.

Base case: If $l = 0$ then x is the empty string so it finishes in $q1$ (starting state).

Induction hypothesis: Suppose that the claim is true for all $l < n$, where $n > 0$.

Induction step: Let $l = n$ and let x' be the first $n - 1$ digits of x. Since the length of x' is less than n the induction hypothesis applies. We have 4 cases:

1. Suppose x' is empty, only contains 0s, or ends with "00" (so one step before the last we are at q1 by the hypothesis). Considering the last input digit, if it is a "0", x still only contains 0s or ends with "00" and indeed M stays at $q1$. If however the input is "1", x ends with "1" and not "101" and indeed M transitions to $q2$ and finishes.

2. Suppose x' ends with "1" and not "101" (so one step before the last we are at $q2$ by the hypothesis) and consider the last input digit. If the input is "0", x ends with "10" and M indeed transitions to $q3$. If however the input is "1", x still ends with "1" and not "101" and indeed M stays at $q2$.

3. Suppose x' ends with "10" (so one step before the last we are at q3 by the hypothesis) and consider the last input digit. If the input is "0", x

ends with "00" and indeed M transitions to $q1$. If however the input is "1", x ends with "101" and indeed M transitions to $q4$.

4. Suppose x' ends with "101" (so one step before the last we are at q4 by the hypothesis) and consider the last input digit. If the input is "0", x ends with "10" and indeed M transitions to $q3$. If however the input is "1", x ends with "1" and not "101" and M indeed transitions to $q2$.

The hypothesis therefore holds for $l = n$ and this completes the proof.

Problem 2

2a

We use contrapositive of *Pumping lemma* to prove regular languages are not closed under tripling. Let $L = \{0, 1\}^*$ be collection of all finite sequence of 0's and 1's. Then, L is regular with the trivial DFA that accepts the start state and remains in the same state.

However, L^3 is not regular. Suppose L^3 is regular with pumping length p. As $0^k1 \in L$, we have

$$w = 0^k10^k10^k1 \in L^3$$

Note that $|w| \geq p$. Let $w = xyz$ be the decomposition satisfying conditions of pumping lemma. In particular, $|xy| \leq p$, $|y| > 0$, and $xy^iz \in L^3$ for all $i \geq 0$. As first p letters of w is only 0, $y = 0^k$ for some $0 < k \leq p$.

Then, $xy^2z = 0^{p+k}10^p10^p1 \notin L^3$. Suppose $xy^2z = uuu$ for some $u \in L$. As then xy^2z contains exactly three 1's, u should contain exactly one 1. Also, u should end same as $uuu = xy^2z$, with the letter 1. By dividing xy^2z right after each 1, we get three substrings which all should be u. If we see the first substring we get $u = 0^{p+k}1$. However, if we see the last substring, we get $u = 0^p$. This gives contradiction as $k > 0$. Thus, L^3 is not regular.

2b

Here, we suppose that the language $L \subseteq \Sigma^*$ is regular. So if L is regular, then there exists a corresponding DFA $M = (Q_M, \Sigma, \delta_M, q_1, F_M)$. Now, we want to find an NFA that describes L^3 to conclude that L^3 is regular. We notice, that if we input a string w into M that lands on an accepting state $q^* \in F_M$, we can link it to an exact copy of M we are going to name M' with an ϵ-transition.

Giving that an alphabet is a unary one, there is only one possible type of transition possible. Thus, we can construct an NFA made of 3 DFAs that are similar to M with ϵ-transitions between the accepting states of the DFAs and the starting states. If w does not hold for L^3, it will not hold for L too so the NFA will fail in the first part of it. However, if it reaches the accepting state of the first automaton, it will reach the overall accepting state.

Refer the following figure

i) Here, we suppose that the language $L \subseteq \Sigma^*$ is regular. So If L is regular, then there exists a corresponding DFA $M = (Q_M, \Sigma, \delta_M, q_1, F_M)$.

Now, we want to find an ~~N~~FA that describes L^3 to conclude that L^3 is regular.

We notice that if we input a string w into M. That lands on ~~the~~ [illegible] an accepting state $q^* \in F_M$, we can link it to ~~the~~ an exact copy of M we going to name M' with an ε- Transition.

Giving that our alphabet is a unary ~~one~~,

There's only one type of transition possible.

Thus, [illegible] we can construct an NFA made of 3 [illegible] same DFA's same as M with ε- transitions between the accepting states of [illegible] of the DFA's and the starting state. If w doesn't hold for $L3$, it won't hold for L to so our NFA will fail in the first part of it. But, if it reaches ~~the~~ accepting state of the first automatony it will reach the overall accepting $\boxed{\text{state}}$

Figure 1: Problem 2b