CMPSC 464: Introduction to the Theory of Computation **Date: 09/07/2022**

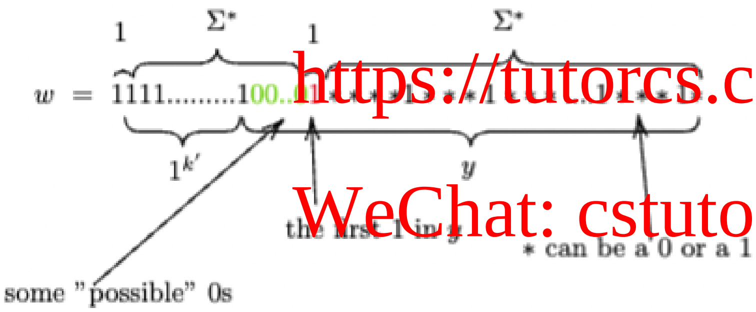
Recitation #2 Solution

Instructor: Dr. Young Kun Ko
TAs: Hamed Mahdavi, Levent Toksoz, Neha Sanjay Rathod, Yuzhang Wang

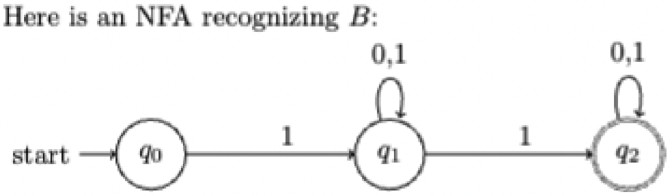
Problem 1

(a) Claim. Every string of form $1\Sigma^*1\Sigma^*$ belongs to B and every string of B can be written as $1\Sigma^*1\Sigma^*$.
proof.
First, notice that every string of form $1\underbrace{\Sigma^*1\Sigma^*}_{y} = 1^1y$, and as you see here $k = 1$ and y actually has AT LEAST one 1. So, every string of that form belongs to B.

Now, we proof that string of B is of form $1\Sigma^*1\Sigma^*$. Let $w \in B$. This means for some $k' \geq 1$, $w = 1^{k'}y'$ such that $y' \in \Sigma^*$ and contains at least k' 1's. The leftmost 1 can be generated by the leftmost 1 in the expression. The rest of the $k' - 1$ of 1s can be generated by some part of the first Σ^*. Now, definitely y can be generated by the rest of the first Σ^* concatenated with $1\Sigma^*$. $\square$



Here is an NFA recognizing B:



(b) We show C is nonregular using the pumping lemma. Assume C is regular and let p be its pumping length. Let $s = 1^p01^p$. The pumping lemma says that $s = xyz$ satisfying the three conditions. Condition three says that y appears among the left-hand 1s. We pump down to obtain the string xz which is not a member of C. Therefore C doesn't satisfy the pumping lemma and hence isn't regular.

Problem 2

Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA recognizing A, where A is some regular language. We construct $M' = (Q', \Sigma, \delta, q_0', F')$ recognizing $NOEXTEND(A)$ as follows:

1. $Q' = Q$
2. $\delta' = \delta$
3. $q_0' = q_0$
4. $F' = \{q | q \in F$ and there is no path of length ≥ 1 from q to an accept state$\}$

If $w \in A$ is a proper prefix of a string in A, then there is a path from the accepting state for w to another accepting state. In this case w will not be accepted by M by definition of F'. If $w \in A$ is not a proper prefix, then there is not a path to another accepting state, so it is still accepted.

Problem 3

Start with the DFA for L, create an NFA by drawing an ε-arrow from the start state to any state that is reachable from the start state. If a string is a suffix, then the NFA can jump to the proper state and read the string, ending at an accept state. If the NFA accepts a string, then it is a suffix since either it is in L, or it can be reached by following the end of an accepting path for a string in L.