CPSC 313: Introduction to Computability **Fall 2016**

Lecture 19: Midterm Review

Lecturer: Dr. Catalin Dohotaru *Scribe: Shane Sims*

19.1 Examples: Regular Expression

Give a regular expression for each of the following languages:

1. The language of all strings containing at least two 0s and at least a 1.

   Solution: The possible patterns of the required symbols are: $001, 010$ and 100. In between we can have any string, $(0+1)^*$. Then any string in this language can be obtained from the regular expression to follow, where each of the three parts correspond to the three possibilities just noted:

   $(0+1)^*0(0+1)^*0(0+1)^*1(0+1)^* + (0+1)^*0(0+1)^*1(0+1)^*0(0+1)^* + (0+1)^*1(0+1)^*0(0+1)^*0$

2. The language of all strings in which the number of 0s and the number of 1s are not both odd.

   Solution: Notice that the requirement that the number of 0s and 1s are not both odd means that either the number of 0s are even, or the number of 1s is even. Then we have the following regular expression:

   $1^*(01^*01^*)^* + 0^*(10^*10^*)^*$

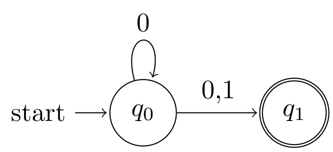
3. The language of all strings in which every run of 0s has a length of at least 3.

   Solution: $(1 + 0000^*)^*)$

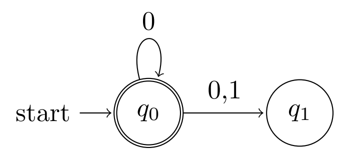
19.2 Examples: Questions concerning NFAs

1. Show that if N is an NFA that recognizes a language L, swapping the accept and non-accept states in N does not necessarily give a new NFA that recognizes $\bar{L}$, as was possible in the case concerning DFAs.

   Solution:



N

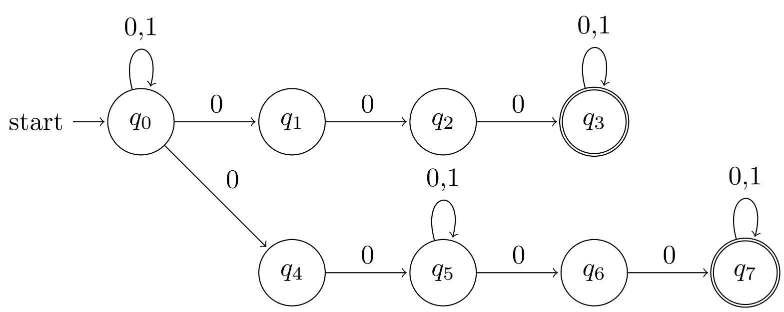


N'

Notice that both NFAs accept 0.

2. Prove that the following language is regular:
   $L = \{w \in \{0,1\} | w$ contains at most one occurrence of the substring $00\}$
   For example, the string 000 has two occurrences of 00.

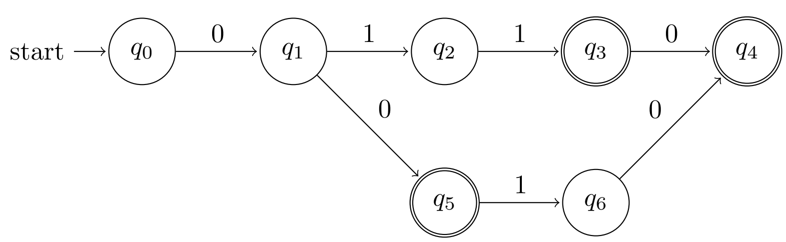
   Solution: The two primary methods we have seen so far to prove that a language is regular, is to provide a finite automation or a regular expression for the language. In this case both of these methods prove difficult. A simpler solution is to make a finite automation (DFA or NFA) for the compliment of L, recalling that once we have done this, we can simply flip the accept and non-accept states, the result being a finite automation for L.
   We observe that $\bar{L}$ =$\{w \in$\{0,1\}*$|w$ contains at least two occurrences of 00\}. We create the following NFA for this language:

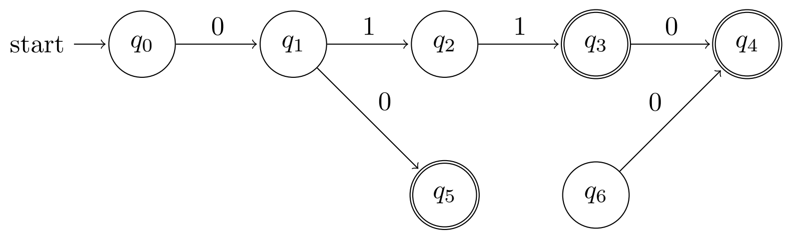


3. Let $L \subseteq \Sigma^*$ be an arbitrary regular language. Prove that the following languages are regular:

   (a) PREFMIN(L) $= \{ xy \in L | x \in L \leftrightarrow y = \epsilon\}$
   (b) SUFMIN(L) $= \{xy \in L | y \in L \leftrightarrow x = \epsilon\}$

   (a) **Solution:** We can reformulate the language as: PREFMIN(L) $= \{ w \in L|$ no proper prefix of w is a member of $L\}$.
   Ex: If $L = \{0110, 00, 0010\}$, then PREFMIN(L) $= \{0110, 00\}$. Notice that 0010 is not a member of the later language because a prefix of this string, namely 00, is already a member of L. We see that example NFAs for L and PREFMIN(L) are respectively as follows:





Now we can see that a NFA recognizing PREFMIN(L), we simply remove all transitions in the DFA for L that originate from a final state.
So if we let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA recognizing L, we construct an NFA, N, recognizing PREFMIN(L): $N = (Q, \Sigma, \delta', q_0, F)$, where

$$\delta'(q, a) = \begin{cases} \delta(q, a) & : q \notin F \\ \emptyset & : q \in F \end{cases}$$

(b) **Solution:** Similarly, we reformulate the language as: SUFMIN(L) = $\{\ w \in L|$ no proper suffix of w is a member of $L\}$.
Ex: If $L = \{10, 010, 100\}$, then SUFMIN(L) = $\{10, 100\}$. Notice that $L^R = \{01, 010, 001\}$, and PREFMIN(L^R) = $\{01, 001\}$.
In general, SUFMIN(L) = (PREFMIN(L^R))^R.
(SUFMIN(L))$^R = \{y^R x^R \in L^R | y^R \in L^R \leftrightarrow x^R = \epsilon\}$.