CS 383 Exam 1 Solutions Fall 2019

There are 6 numbered questions. The 8 parts of Question 1 are worth 3 points each. Questions 2 through 6 are worth 15 points each. You get one point for free.

- 1. Which languages are regular? You don't need to prove your answers. Write an "R" in the blank next to the description of each language you think is regular. Write "N" for any language you think is not regular. In each case the alphabet is Σ ={0,1}
 - a. ____R___Strings of 0s and 1s that start and end on the same digit, such as 10010101 and 010. = 0(0+1)*0 + 1(0+1)*1
 - b. __N_Strings with odd length that have 0 as the center element, such as 101 or 01000
 - c. ____R_ Strings of 0s, 1s and 2s whose digits sum to more than 9. For example, 21202201, whose digits sum to 10, is such a string.
 - d. ___R_Strings whose length is divisible by 5, such as $0^{3}1^{7}$.
 - e. __N_Strings whose length is a perfect square, such as 0^91^7 .
 - f. __N_ { $0^{j}0^{k}1^{k} | j \ge 0, k \ge 0$ }
 - g. __R_ { $0^i0^k1^k1^n$ | i>=0, k>=0, n>=0 } = 0^*1^*
 - h. ____R_ { $0^k0^k | k \ge 0$ } =strings of 0s with even length

2. Give a DFA for the strings of 0s and 1s that **do not** contain 010 as a substring.

If you can make a DFA for the strings that DO contain 010 and this DFA has all possible transitions from each node, just invert the final states:



- 3. Here is an ϵ -NFA, with start state A.
 - a) Convert this NFA to a DFA
 - b) Give a regular expression for the strings it accepts.



Here's my DFA:



Regular expression: $0^++0^*10^+$

4. Suppose \mathcal{L}^* is regular. Must \mathcal{L} be regular? Either prove that it must be or give an example to show it ain't necessarily so.

No. If \mathcal{L} is any language that contains both 0 and 1 then \mathcal{L}^* contains all strings of 0s and 1s, so \mathcal{L}^* is regular. For the example we just need any language that contains 0 and 1 and isn't regular, such as $\mathcal{L} = \{0^n 1^n \mid n \ge 0\} \cup \{0,1\}$. The usual pumping lemma argument shows \mathcal{L} is not regular, but \mathcal{L}^* is denoted by the regular expression (0+1)*.

5. Consider the following DFA. We had an algorithm for converting a DFA to a regular expression. This involved making a table of regular expressions r_{ij}^k .



Here is the first column of a table of the r_{ij}^k expressions; find the 4 entries of the second column.

	k=0	k=1
r_{11}^{k}	ε+1	1*
r_{12}^{k}	0	1*0
r_{21}^k	0	01*
r_{22}^k	ε+1	=ɛ+1+01*0

```
r_{ij}^{1} = r_{ij}^{0} + r_{i1}^{0} (r_{11}^{0})^{*} r_{1j}^{0}
r_{11}^{1} = \varepsilon + 1 + (\varepsilon + 1)1^{*}(\varepsilon + 1) = 1^{*}
r_{12}^{1} = 0 + (\varepsilon + 1)1^{*}0 = 1^{*}0
r_{21}^{1} = 0 + 01^{*}(\varepsilon + 1) = 01^{*}
r_{22}^{1} = \varepsilon + 1 + 01^{*}0
```

6. Consider the language of even-length strings of 0s and 1s whose first half is all 0s. For example, 000101, 000011 and even 000000 are all strings in this language. Use the Pumping Lemma to show carefully that this language is not regular.

If this language is regular, let p be its pumping constant. Consider the string $w=0^{p}1^{p}$, which is in this language. I will show that w is not pumpable. Let w=xyz be any decomposition of w into 3 parts with y not empty and |xy| <= p. Since the first p letters of w are all 0, y must consist of a positive number of 0s. The string xz has fewer 0s than 1s, so either it has odd length or its first half contains some 1s. Either way, xz is not in the language. Since there is no decomposition of w that can be pumped, this language can't be regular.