CS/ECE 374 FALL 2018	Zhe Zhang (zzhan157@illinois.edu)
Homework 1 Problem 1	Ray Ying (xinruiy2@illinois.edu)
	Anqi Yao (anqiyao2@illinois.edu)

1.For each of the following languages over the alphabet $\{0, 1\}$, give a regular expression that describes that language, and briefly argue why your expression is correct.

---

1. All strings except 010.

   Solution: The string with length zero is ϵ
   The strings with length one are $0, 1$
   The strings with length two are $00, 01, 10, 11$
   The strings with length three except 010 are $000, 001, 011, 100, 101, 110, 111$
   The regular expression $(1+0)(1+0)(1+0)(1+0)(1+0)^*$ refers to the strings with length at least four
   Thus the regular expression that describes all strings except 010 is
   $\epsilon + 0 + 1 + 00 + 01 + 10 + 11 + 000 + 001 + 011 + 100 + 101 + 111 + (1+0)(1+0)(1+0)(1+0)(1+0)^*$ ■

2. All strings that end in 10 and contain 101 as a substring.

   Solution: There are two possible situations:

   (a) 101 is in the middle of the string $(0+1)^*101(0+1)^*10$

   (b) 101 is at the end so that it belongs last four digits 1010. There exists a 0 following the substring 101 since the string should end in 10. $(0+1)^*1010$

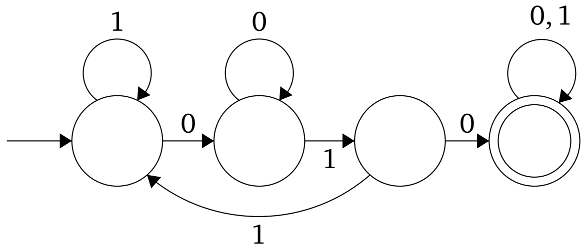
   Thus the regular expression that describes all strings that end in 10 and contain 101 is
   $(0+1)^*101(0+1)^*10 + (0+1)^*1010$ ■

3. All strings in which every nonempty maximal substring of 1s is of length divisible by 3. For instance 0110 and 101110 are not in the language while 11101111110 is.

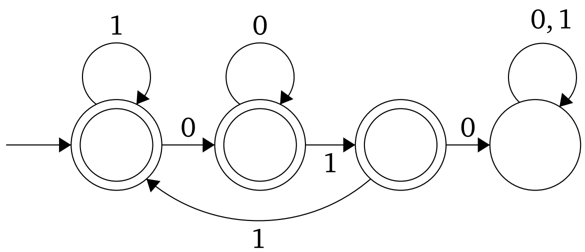
   Solution: Since every nonempty maximal substring of 1 is of length divisible by 3, we could use $(111)^*$ to represent every occurrence of 1s.
   Thus the regular expression is $(0+111)^*$ ■

4. All strings that do not contain the substring 010.

   Solution: Let L be the language represents all strings that contain the substring 010.
   We could draw a DFA that accepts the language L.



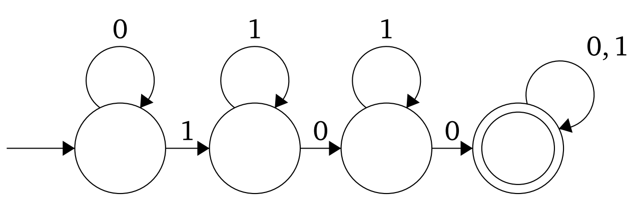
L^c is the complement of the language L and L^c represents all strings that do not contain the substring 010
We could draw a DFA that accepts the language L^c (simply change the non-accepting states to accepting states and vice versa)



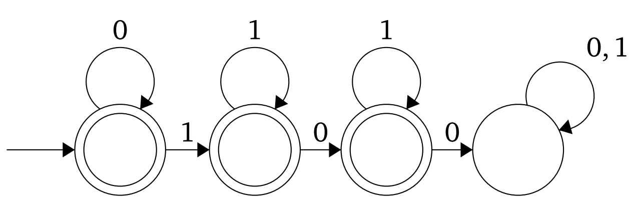
Then we transform the DFA to the regular expression:
$(\mathtt{1}+\mathtt{0}^*\mathtt{11})^* + (\mathtt{1}+\mathtt{0}^*\mathtt{11})^*\mathtt{00}^* + (\mathtt{1}+\mathtt{0}^*\mathtt{11})^*\mathtt{00}^*\mathtt{1} = (\mathtt{1}+\mathtt{0}^*\mathtt{11})^*(\epsilon + \mathtt{00}^* + \mathtt{00}^*\mathtt{1})$ ■

5. All strings that do not contain the *subsequence* $\mathtt{100}$.

Solution: Let L be the language represents all strings that contain the subsequence 100. We could draw a DFA that accepts the language L.



L^c is the complement of the language L and L^c represents all strings that do not contain the subsequence 100
We could draw a DFA that accepts the language L^c (simply change the non-accepting states to accepting states and vice versa)



Then we transform the DFA to the regular expression:
$\mathtt{0}^* + \mathtt{0}^*\mathtt{11}^* + \mathtt{0}^*\mathtt{11}^*\mathtt{01}^* = \mathtt{0}^*(\epsilon + \mathtt{11}^* + \mathtt{11}^*\mathtt{01}^*) = \mathtt{0}^*(\mathtt{1}^*(\epsilon + \mathtt{0}))\mathtt{1}^*$ ■