CS/ECE 374 Spring 2017
Homework 1 Problem 1

Lanxiao Bai (lbai5@illinois.edu)
Renheng Ruan (rruan2@illinois.edu)

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 **101**.
2. All strings that end in **01** and contain **000** as a substring.
3. All strings in which every nonempty maximal substring of **0**s is of odd length. For instance **1001** is not in the language while **0100010** is.
4. All strings that do not contain the substring **101**.
5. All strings that do not contain the subsequence **101**.

Solution:

1. $(0+11+100+101(0+1))(0+1)^* + \varepsilon + 1 + 10$

   Reason: Empty string is accepted, all strings start with **0, 11, 100** are accepted, strings with length greater than 4 are accepted, **1** and **10**, although not start with the prefix above, are accepted.

2. $(0+1)^*000(1+(0+1)^*01)$

   Reason: String can start with everything, but must contain **000**, which can immediately ends with **1** to form **01** tail or has any substring after that with a **01** ending.

3. $(0+1)^*1(00)^*01(0+1)^*$

   Reason: Since what we concern about is the substring, so basically the start and end any be any string formed under alphabet. And the consecutive **0**s of odd length and be formed by $(00)^*0$. Since the question requires it to be maxima substring, **1**s on the both sides are required to separate the substring from the rest part.

4. $0^*(1(\varepsilon+000^*)1)^*0^*$

   Reason: The main idea is that, in the string, every 2 **1**s has 0 or more than 2 **0**s between them. Also, the string can start or end with **0**.

5. $0^*1^*0^*$

   Reason: The main idea is that in the string, there should be nothing or **1**s only between 2 **1**s and string can start and/or end with **0**.

■

CS/ECE 374 Spring 2017	Lanxiao Bai (lbai5@illinois.edu)
Homework 1 Problem 2	Renheng Ruan (rruan2@illinois.edu)

Let L be the set of all strings in $\{0, 1\}^*$ that contain at most two occurrences of the substring 100.

1. Describe a DFA that over the alphabet $\Sigma = \{0, 1\}$ that accepts the language L. Argue that your machine accepts every string in L and nothing else, by explaining what each state in your DFA *means*.

   You may either draw the DFA or describe it formally, but the states Q, the start state s, the accepting states A, and the transition function δ must be clearly specified.

2. Give a regular expression for L, and briefly argue why the expression is correct.

Solution: 1. The DFA $M = \{Q, \Sigma, \delta, s, A\}$ accepts L should look like the automata below, with state $q \in Q$ in form of (i, j) that i denotes the number of received 100 and j denotes the constructing 100.

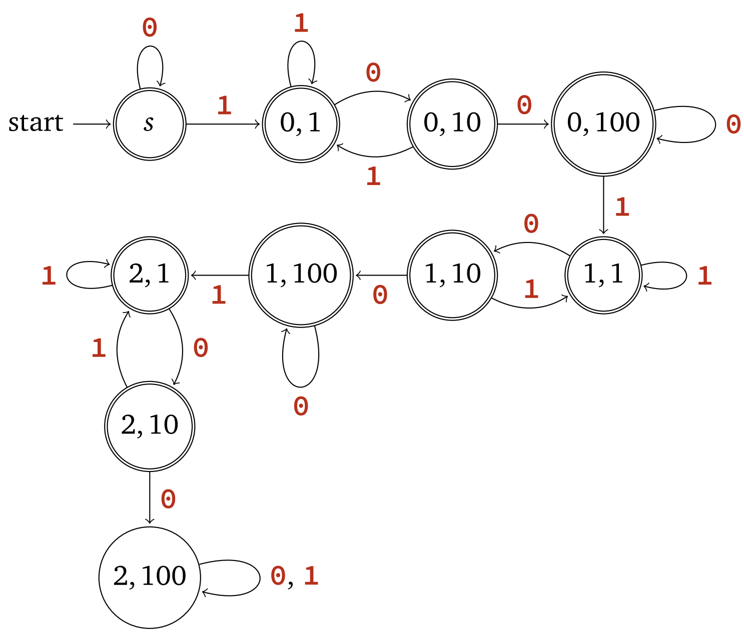

Figure 1. DFA that accepts L

2. $(0^*(\varepsilon + 1)(\varepsilon + 0))^*(100)(0^*(\varepsilon + 1)(\varepsilon + 0))^*(100)(0^*(\varepsilon + 1)(\varepsilon + 0))^*$

   Reason: Other than two 100s, the other parts, if have 1, should have no more than 2 consecutive 0s.

■

CS/ECE 374 Spring 2017 Lanxiao Bai (lbai5@illinois.edu)
Homework 1 Problem 3 Renheng Ruan (rruan2@illinois.edu)

Let L_1, L_2, and L_3 be regular languages over Σ accepted by DFAs $M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1)$, $M_2 = (Q_2, \Sigma, \delta_2, s_1, A_2)$, and $M_3 = (Q_3, \Sigma, \delta_3, s_3, A_3)$, respectively.

1. Describe a DFA $M = (Q, \Sigma, \delta, s, A)$ in terms of M_1, M_2, and M_3 that accepts $L = \{w \mid w \text{ is in exactly two of } \{L_1, L_2, L_3\}\}$. Formally specify the components Q, δ, s, and A for M in terms of the components of M_1, M_2, and M_3.
2. Prove by induction that your construction is correct.

Solution: 1. For $M = (Q, \Sigma, \delta, s, A)$ that accepts exactly two of $\{L_1, L_2, L_3\}$,

We require that:

- $Q = Q_1 \times Q_2 \times Q_3$
- Σ to be the same as M_1, M_2, M_3
- $\delta : Q \times \Sigma \to Q$, that $\delta((q_1, q_2, q_3), a) = (\delta_1(q_1, a)), \delta_2(q_2, a), \delta_3(q_3, a)), a \in \Sigma, q_1 \in Q_1, q_2 \in Q_2, q_3 \in Q_3$
- $s = (s_1, s_2, s_3)$
- $A = (A_1 \times A_2 \times (Q_3 - A_3)) \cup (A_1 \times (Q_2 - A_2) \times A_3) \cup ((Q_1 - A_1) \times A_2 \times A_3)$

2. **Proof:** Apply induction on the length of w.

Base case: When $|w| = 0$, namely, $w = \varepsilon$, $\delta(s, w) = \delta(s, \varepsilon) = (\delta_1(s_1, \varepsilon), \delta_2(s_2, \varepsilon), \delta_3(s_3, \varepsilon)) = (s_1, s_2, s_3) = s$.

If $w \in L$, without losing generality we can assume $w \in L_1, L_2$ only, so $\delta_1(s_1, w) = s_1 \in A_1, \delta_2(s_2, w) = s_2 \in A_2, \delta_3(s_3, w) = s_3 \notin A_3$. Thus, $s \in A$, which means M accepts w.

If M accepts w, then $s = (s_1, s_2, s_3) \in A$, which means that $s_1 \in A_1, s_2 \in A_2, s_3 \notin A_3$ or $s_1 \in A_1, s_2 \notin A_2, s_3 \in A_3$ or $s_1 \notin A_1, s_2 \in A_2, s_3 \in A_3$. Then exactly 2 of M_1, M_2, M_3 accepts w, so w in exactly 2 of L_1, L_2, L_3. Hence, $w \in L$.

Suppose for all $|w| \leq k$, if $w \in L$, then M accepts w. And if M accepts w, then $w \in L$.

Then when $|w| = k + 1$, let $w = w_0 \bullet a, a \in \Sigma$, $\delta(s, w) = \delta(\delta(s, a), w_0)$. By induction hypothesis, we know that $\delta(s, a)$ corrected judged if a should be accepted. And let $\delta(s, a) = q$, $\delta(q, w_0)$, since $|w_0| = k$, again, by induction hypothesis, we know that $\delta(q, w_0)$ corrected judged if w_0 should be accepted by M.

If M accepts w, then $\delta(s, w) \in A \Rightarrow (\delta_1(s_1, w), \delta_2(s_2, w), \delta_3(s_3, w)) \in A$. That means that $\delta_1(s_1, w) \in A_1, \delta_2(s_2, w) \in A_2, \delta_3(s_3, w) \notin A_3$ or $\delta_1(s_1, w) \in A_1, \delta_2(s_2, w) \notin A_2, \delta_3(s_3, w) \in A_3$ or $\delta_1(s_1, w) \notin A_1, \delta_2(s_2, w) \in A_2, \delta_3(s_3, w) \in A_3$, so exactly 2 of M_1, M_2, M_3 accepts w, so w in exactly 2 of L_1, L_2, L_3. Hence $w \in L$.

In conclusion, the automata M described above is correct.

■