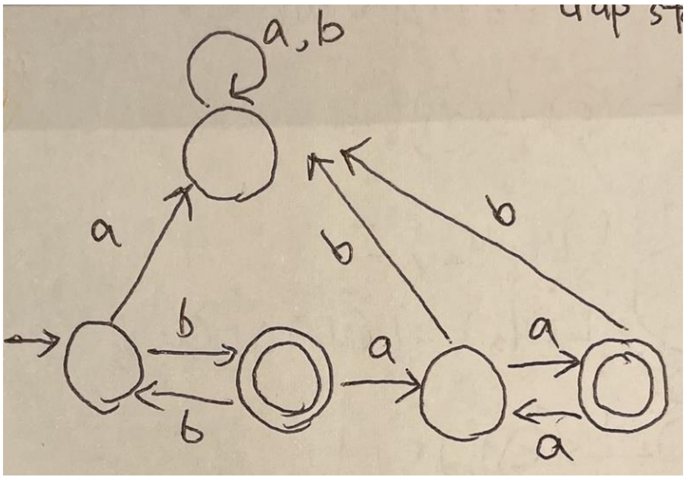
자동장치이론 HW1

20181593 계인혜

1.12

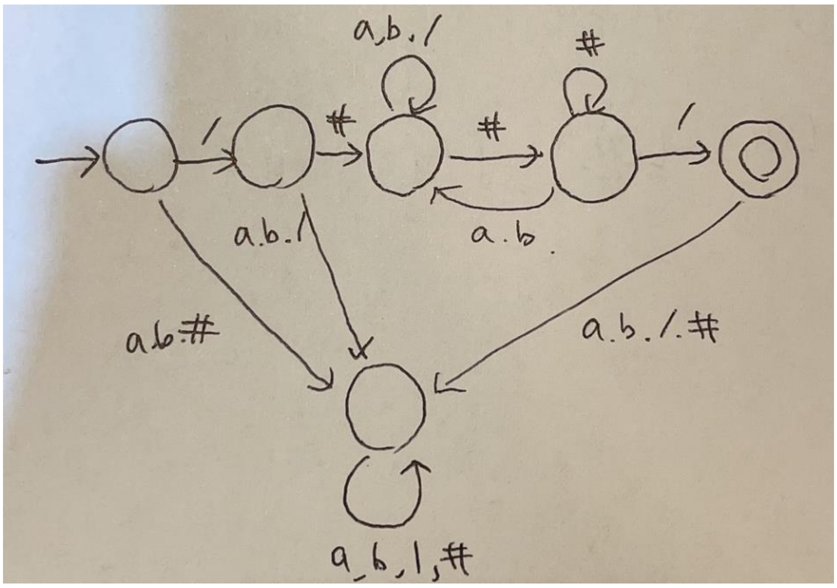
a. DFA :



b. Regular expression : b(bb)*(aa)*

1.22

a. DFA :



b. Regular expression : /#(a∪b∪/∪(#*(a∪b)))*#/

1.30

Example 1.73 :

Let p be the pumping length and choose s to be the string 0^p1^p. You know that s is a member of 0*1* and has length at least p. |xy| must be ≤ p, so y consists of all zeros. Let i=0, so then the string xz contains fewer zeros and therefore the string s cannot be pumped. Thus, we have a contradiction, so 0*1* is not regular.

Error : The string xz does contain fewer zeros. However, the resulting string is still in the language because 0*1* allows an arbitrary number of zeros. In other words, the language 0*1* allows string with different number of 0's and 1's. Hence, there is no contradiction.

1.31

Let $D_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$, $D_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ be two DFAs that recognize A and B, respectively. Here, we shall construct a DFA $D = (Q, \Sigma, \delta, q, F)$, that recognizes the perfect shuffle of A and B.

At first, DFA D can be defined as follows.

- Q = $Q_A \times Q_B \times \{A, B\}$, which keeps track of all possible current states of two DFAs and which DFA to match.

- q = (q_A, q_B, A), which states that D starts with D_A, D_B and the next character should be in D_A.

- F = $F_A \times F_A \times \{A\}$, which states that D accepts the string if both $D_A\ and\ D_B$ are in accept states, and the next character should be in D_A.

And we have to make D switch from running D_A and running D_B after each character is read. So D needs to keep track of the current state of D_A and D_B.

- $\delta = \delta\big((x, y, A), a\big) = (\delta_A(x, a), y, B)$, which x is current state of D_A, y is current state of D_B, and next character is a. We should change the current state of A to $\delta_A(x, a)$, B's current state is same, and the next character will be in D_B.

Plus, we have to keep track of whether the next character should be matched in D_A and D_B.

- $\delta = \delta((x, y, B), b) = (x, \delta_B(y, b), A)$, which x is current state of D_A, y is current state of D_B, and next character is b. We should change the current state of B to $\delta_B(y, b)$, A's current state is same, and the next character will be in D_A.

1.40

The proof is by contradiction. Assume that E is a regular language. By the pumping Lemma, there is a constant p associated with E.

At first, choose the string $s = \begin{bmatrix}1\\0\end{bmatrix}^p \begin{bmatrix}0\\1\end{bmatrix}^p$. $s \in L$ because $1^p0^p = (0^p1^p)^R$, and |s| = 2p $\geq$ p.

Then, s can be written as s = xyz such that $x = \varepsilon,\ y = \begin{bmatrix}1\\0\end{bmatrix}^p,\ z = \begin{bmatrix}0\\1\end{bmatrix}^p, and\ |y| = p \geq p.$

In any possible division y = uvw, we mush have $v = \begin{bmatrix}1\\0\end{bmatrix}^m, where\ 0 < m \leq p.$

Choose i=2, then $xyz = xuv^2wz = \begin{bmatrix}1\\0\end{bmatrix}^{p+m} \begin{bmatrix}0\\1\end{bmatrix}^p$. Because $m > 0,\ 1^{p+m}0^p \neq (0^{p+m}1^p)^R$, that is the top row can't be the reverse of the bottom row. Thus, $xuv^2wz \notin L$. This is a contradiction. Hence, L is not a regular language.

1.52

Suppose that Y is regular. Let p be the pumping length given by pumping lemma. Let s = $1^p \# 1^{p+1} \# \dots \# 1^{2p}$. Because $s \in Y$ and $|s| \geq p$, we can split s into xyz such that for $i \geq 0$, $xy^iz \in Y$ by the pumping lemma.

Because $|xy| \leq p\ and\ |y| \geq 0, y$ must consist of between 1 and p ones. The string xyyz would generate between p+1 and 2p of 1s before the first #. In each case, $s \notin Y$ because some $x_i \neq x_i\ ,\ for\ i \neq j$. This violates the pumping lemma, therefore Y is not regular.

1.38

For a top row with n bits and a bottom row with n bits, denote them as $t = t_{n-1}\, t_{n-2} \dots t_1\, t_0$,

B= $b_{n-1}\, b \dots b_1\, b_0$ separately. Read w in reverse order because for any language A, if A is regular then the reverse of A is also regular. In order to be the bottom row is three times the top row, the following conditions must be satisfied where $c_i\ is\ carry\ bit,\ t_i\ and\ b_i\ is\ LSB,\ t_{-1} = c_0 = 0\ and$

$a \oplus b = (a \wedge b) \vee (a \vee b)$.

- $b_i = t_i \oplus t_{i-1} \oplus c_i$
- $b_i = t_i$
- $c_i = (t_i \wedge t_{i-1}) \vee (t_i \vee t_{i-1}) \wedge c_i$

Now we need to construct DFA M with 4 states, and a trap state.

$q_0 : \{c_i, t_{i-1}\} = \{0,0\}$

$q_1 : \{c_i, t_{i-1}\} = \{0,1\}$

$q_2 : \{c_i, t_{i-1}\} = \{1,1\}$

$q_3 : \{c_i, t_{i-1}\} = \{1,0\}$