Homework 2

Question 1:

Given $w \in \Sigma^*$, we can represent the number in the first row of w as w_{up}, the number of the second row as w_{dw}. Then we can define D as $\forall w \in D,\ w_{up} > w_{dw}$ (**Def of D**)

To show D is regular, we can construct an DFA $M = (\Sigma, Q, s, A, \delta)$ that accept strings in D.

$$
\begin{aligned}
&\Sigma = \{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}\} \\
&Q = \{q | q \in \mathbb{R}\} \\
&A = \{q \in Q | q > 0\} \\
&s = 0 \\
&\delta : \Sigma \to \mathbb{R}, \\
&\delta(a) = a_{up} - a_{dw} = \begin{cases} 1 & \text{,if a=} \begin{bmatrix}1\\0\end{bmatrix} \\ -1 & \text{,if a=} \begin{bmatrix}0\\1\end{bmatrix} \\ 0 & \text{,otherwise} \end{cases}
\end{aligned}
\tag{1}
$$

We use $w[i]$ to represent the i-th digit in w, counting from left to right. Since we need to compare two rows' value digit by digit, and the i-th digit is 2 times larger than the $(i+1)$-th digit, we construct our extended transition function as:

$$
\delta^*(w) = \begin{cases} 2^{|x|}(\delta(a)) + \delta^*(x) & \text{,if } w = a \cdot x \\ 0 & \text{,if } w = \epsilon \end{cases}
\tag{2}
$$

Since any binary number $a_1 a_2 ... a_n = \sum_i 2^{n-i} a_i$ (we denote this summation as **Bi_sum**), and $\sum_{i=1}^{|\epsilon|} x = \sum_{i=1}^{0} x = 0$ (one of properties of summation), we can merge and fully expand $\delta^*(w)$ as:

$$
\delta^*(w) = \sum_{i=1}^{|w|} 2^{|w|-i}(\delta(w[i]))
\tag{3}
$$

Proof $\forall w \in \Sigma^*, \delta^*(w) \in A \iff w \in D$

- $\Rightarrow \delta^*(w) \in A$. Then we have:

$$
\begin{aligned}
\delta^*(w) &= \textstyle\sum_{i=1}^{|w|} 2^{|w|-i}(\delta(w[i])) && \text{Def of } \delta^* \\
&= \textstyle\sum_{i=1}^{|w|} 2^{|w|-i}(w[i]_{up} - w[i]_{dw}) && \text{Def of } \delta \\
&= \textstyle\sum_{i=1}^{|w|} 2^{|w|-i}(w[i]_{up}) - \sum_{i=1}^{|w|} 2^{|w|-i}(w[i]_{dw}) && \text{Sumation Property} \\
&= w_{up} - w_{dw} && \text{Def of Bi\_sum}
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
\delta^*(w) \in A &\Rightarrow (w_{up} - w_{dw}) > 0 && \text{Def of A} \\
&\Rightarrow w_{up} > w_{dw} \\
&\Rightarrow w \in D && \text{Def of D}
\end{aligned}
$$

Thus if $\delta^*(w) \in A \Rightarrow w \in D$

- $\Leftarrow w \in D$:

$$\begin{aligned}
w \in D \quad &\Rightarrow w_{up} > w_{dw} && \text{Def of D} \\
&\Rightarrow w_{up} - w_{dw} > 0 \\
&\Rightarrow \sum_{i=1}^{|w|} 2^{|w|-i}(w[i]_{up}) - \sum_{i=1}^{|w|} 2^{|w|-i}(w[i]_{dw}) > 0 && \text{Def of Bi_sum} \\
&\Rightarrow \sum_{i=1}^{|w|} 2^{|w|-i}(w[i]_{up} - w[i]_{dw}) > 0 && \text{Sumation Property} \\
&\Rightarrow \sum_{i=1}^{|w|} 2^{|w|-i}(\delta(w[i])) > 0 && \text{Def of } \delta \\
&\Rightarrow \delta^*(w) > 0 && \text{Def of } \delta^* \\
&\Rightarrow \delta^*(w) \in A && \text{Def of A}
\end{aligned} \tag{5}$$

Thus it is true that $w \in D \Rightarrow \delta^*(w) \in A$

Therefore, it is true that $\forall w \in \Sigma^*, \delta^*(w) \in A \iff w \in D$. So D can be accepted by DFA $M = (\Sigma, Q, s, A, \delta)$. According to Kleene's Theorem, D is regular.

Question 2

- **Part a**:

  We use (w, s) to represent the current output string w and current state s, the new output character will be added to the right side of w. The character shown above the arrow is input character, the transition rule applied to the input is shown below the arrow. Then we can have:

  - input **aaca**:

$$(\epsilon, n_0) \xrightarrow[a:b]{a} (b, n_0) \xrightarrow[a:b]{a} (bb, n_0) \xrightarrow[c:a]{c} (bba, n_1) \xrightarrow[a:a]{a} (bbaa, n_1) \tag{6}$$

  - input **cbbc**:

$$(\epsilon, n_0) \xrightarrow[c:a]{c} (a, n_1) \xrightarrow[b:b]{b} (ab, n_0) \xrightarrow[b:c]{b} (abc, n_0) \xrightarrow[c:a]{c} (abca, n_1) \tag{7}$$

  - input **bcba**:

$$(\epsilon, n_0) \xrightarrow[b:c]{b} (c, n_0) \xrightarrow[c:a]{c} (ca, n_1) \xrightarrow[b:b]{b} (cab, n_0) \xrightarrow[a:b]{a} (cabb, n_0) \tag{8}$$

  - input **acbbca**:

$$(\epsilon, n_0) \xrightarrow[a:b]{a} (b, n_0) \xrightarrow[c:a]{c} (ba, n_1) \xrightarrow[b:b]{b} (bab, n_0) \xrightarrow[b:c]{b} (babc, n_0) \xrightarrow[c:a]{c} (babca, n_1) \xrightarrow[a:a]{a} (babcaa, n_1) \quad (9$$

- **Part b**:

$$FST_1 = (\Sigma_1, \Gamma_1, Q_1, s, \delta_1)$$

$$\begin{cases} \Sigma_1: \text{ input alphabet} \\ \Gamma_1: \text{ output alphabet} \\ Q_1: \text{ set of states} \\ s: \text{ starting state, } s \in Q \\ \delta_1 : Q_1 \times \Sigma_1 \to Q_1 \times \Gamma_1 \end{cases} \tag{10}$$

- **Part c**:

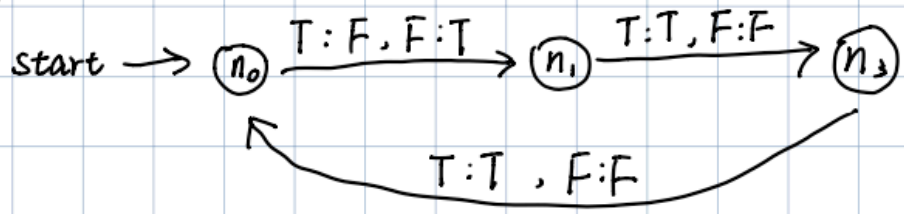
  For FST_0, we set the input alphabet as Σ_0, output alphabet as Γ_0, set of state as Q_0, the initial state as s

$$FST_0 = (\Sigma_0, \Gamma_0, Q_0, s, \delta_0)$$

$$\Rightarrow \begin{cases} \Sigma_0 = \{a, b, c\} \\ \Gamma_0 = \{a, b, c\} \\ Q_0 = \{n_0, n_1\} \\ s = n_0 \\ \delta_0 : Q_0 \times \Sigma_0 \to Q_0 \times \Gamma_0 \\ \delta_0(q, x) = \begin{cases} (n_0, b) & \text{, if } (q, x) = (n_0, a) \\ (n_0, c) & \text{, if } (q, x) = (n_0, b) \\ (n_1, a) & \text{, if } (q, x) = (n_0, c) \\ (n_1, a) & \text{, if } (q, x) = (n_1, a) \\ (n_1, c) & \text{, if } (q, x) = (n_1, c) \\ (n_0, b) & \text{, if } (q, x) = (n_1, b) \end{cases} \end{cases} \tag{11}$$

- **Part d:**

(d)



There are three states because it helps me find indices divisible by 3, and n_0 is the invert state. Other states will not change the character.

Question 3

- **Part a:**

3\. (a) Let $M = (\Sigma, Q, s, A, \delta)$ be a DFA that accepts L.
We construct an NFA M' that reads its input string w and simulates the original DFA M reading the input string ww^R, and it accepts palin (L).
Let $h = \delta^*(s, w)$ be the state that M will reach after reading input w.
M' includes a special start state s' with ε-transitions to every double of the form (s, q), where $q \in A$.

Define $M' = (\Sigma, Q', s, A', \delta')$ as follows.

$$Q' = (Q \times Q) \cup \{s'\}$$
$$A' = \{(h, h) \mid h \in Q\}$$
$$\delta'(s', \varepsilon) = \{(s, q) \mid q \in A\}$$
$$\delta'(s', a) = \emptyset \qquad \forall a \in \Sigma$$
$$\delta'((p, q), \varepsilon) = \emptyset \qquad \forall p, q \in Q$$
$$\delta'((p, q), a) = \{(\delta(p, a), q_1) \mid q \in \delta(q_1, a)\} \qquad \forall p, q \in Q \text{ and } a \in \Sigma$$

Because palin(L) has an NFA M', so palin(L) is regular.

Because the length of w and w^R are the same, and we can start from the start state of M and the accepting state of M simultaneously to find the middle state of M. Thus, we use the cross product of two states in M as the states in M'. The start state of the first state of Q' is the start state of M and it goes to the middle state of M. And the start state of the second state of Q' is the accepting state of M and it goes to the middle state as well. The cross product of the two same middle states is the accepting state of M'.

- **Part b:**

(b) Let $M = (\Sigma, Q, s, A, \delta)$ be a DFA that accepts L.
We construct an NFA M' that reads its input string y and simulates the original DFA M reading the input string yx, and it accepts cycle(L).
Let $h = \delta^*(s, y)$ be the state that M will reach after reading input y.
M' includes a special start state s'' with ε-transitions to h in M.

Define $M' = (\Sigma, Q', s', A', \delta')$ as follows.

$Q' = Q \times \{s''\}$

$s' = h$	$\forall h \in Q$
$A' = h$	$\forall h \in Q$
$\delta'(s'', \varepsilon) = h$	$\forall h \in Q$
$\delta'(s'', a) = \emptyset$	$\forall a \in \Sigma$
$\delta'(q, \varepsilon) = \emptyset$	$\forall q \in Q \setminus A$
$\delta'(q, \varepsilon) = p$	$\forall q \in A$ and $p \in S$
$\delta'(q, a) = \emptyset$	$\forall q \in A$ and $a \in \Sigma$
$\delta'(q, a) = \delta(q, a)$	$\forall q \in Q \setminus A$ and $a \in \Sigma$

Because cycle(L) has an NFA M', so cycle(L) is regular.

To distinguish x and y, we need to the break point between them, and h is the break point. And then we need to switch their positions. After that, the start state and end state of M' are both the break point. Then, the break point of M' is from the accepting state of M to the start state of M. Thus, for the transition function, when we find the accepting state of M, we need to take a free step to reach the start state of M. That's how yx becomes xy.

- **Part c:**

(c) Let $M = (\Sigma, Q, s, A, \delta)$ be a DFA that accepts L.
We construct an NFA M' that reads its input string x and simulates the original DFA M reading the input string y, and it accepts subseq(L).
M' includes a special start state s' with ε-transitions to A in M.

Define $M' = (\Sigma, Q', s, A', \delta')$ as follows.

$Q' = Q \times \{s'\}$

$A' = A$

$\delta'(s', \varepsilon) = A$

$\delta'(s', a) = \emptyset \qquad \forall a \in \Sigma$

$\delta'(q, \varepsilon) = \delta(q, a) \qquad \forall a \in \Sigma$ and $q \in Q$

$\delta'(q, a) = \delta(q, a)$ [redacted]

Because subseq(L) has an NFA M', so subseq(L) is regular.

$\{\delta(q,a) \mid \forall a \in \Sigma\}$

To use the subsequence in M' to simulate the full input in M, we need to skip some characters. Thus, we have transition function $\delta'(q, \varepsilon) = \delta(q, a)$, then we can have free steps to skip some characters.