Question 1:

We define $\hat{\delta}$ as $\hat{\delta} : Q \times \Sigma^* \to Q$ inductively as:

$$\hat{\delta}(q, \epsilon) = q \quad \text{for all } q \in Q \tag{1}$$

$$\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a) \quad \text{for all } q \in Q, x \in \Sigma^*.a \in \Sigma \tag{2}$$

$$\begin{aligned} a. \quad \hat{\delta}(p, a) &= \hat{\delta}(p, \epsilon a) \\ &= \delta(\hat{\delta}(p, \epsilon), a) \quad \text{by definition (2)} \\ &= \delta(p, a) \quad \text{by definition (1)} \end{aligned}$$

b. We define δ_C to be the transition function for product construction such that

$$\delta_C((p, q), a) = (\delta_A(p, a), \delta_B(q, a)) \quad \text{for all a} \in \Sigma \tag{3}$$

Let P(n) be the statement $\hat{\delta}_C((p, q), x) = (\hat{\delta}_A(p, x), \hat{\delta}_B(q, x)) \quad \text{for all } x \in \Sigma^*$ with length n . We want to show P(n) is true for all $n \geq 0$

Base case: we want to show P(n) is true for n = 0
By definiton(1), we have

$$\hat{\delta}_C((p, q), \epsilon) = (p, q) = (\hat{\delta}_A(p, \epsilon), \hat{\delta}_B(q, \epsilon))$$

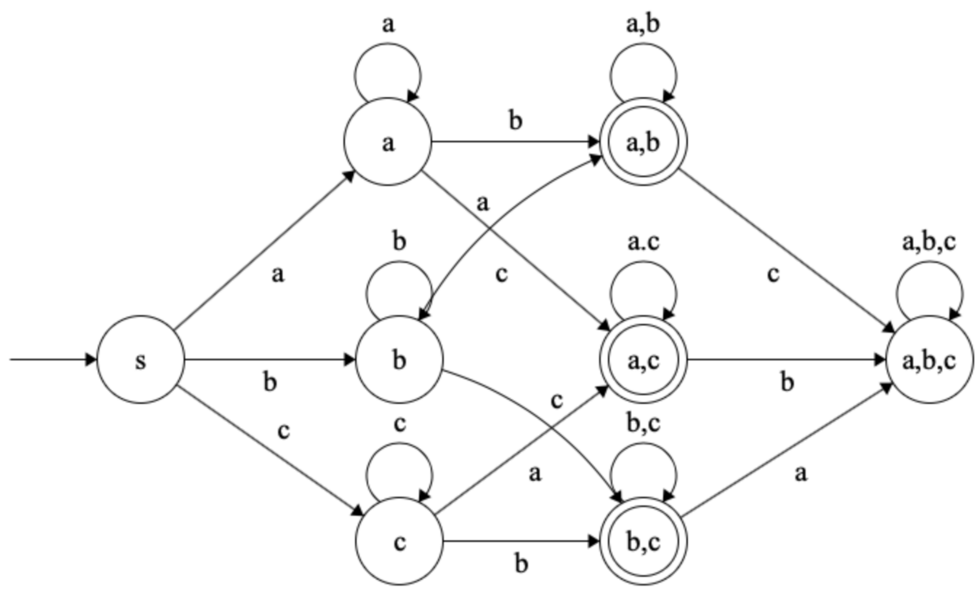
Induction Hypothesis: Assume P(n) is true for $n < k$, where k is a fixed integer, that is
$\hat{\delta}_C((p, q), x) = (\hat{\delta}_A(p, x), \hat{\delta}_B(q, x)) \quad$ for all $x \in \Sigma^*$ with length less than k .

Induction Step: we want to prove P(k), that is,
$\hat{\delta}_C((p,q),y) = (\hat{\delta}_A(p,y), \hat{\delta}_B(q,y))$ for all $y \in \Sigma^*$ with length k .

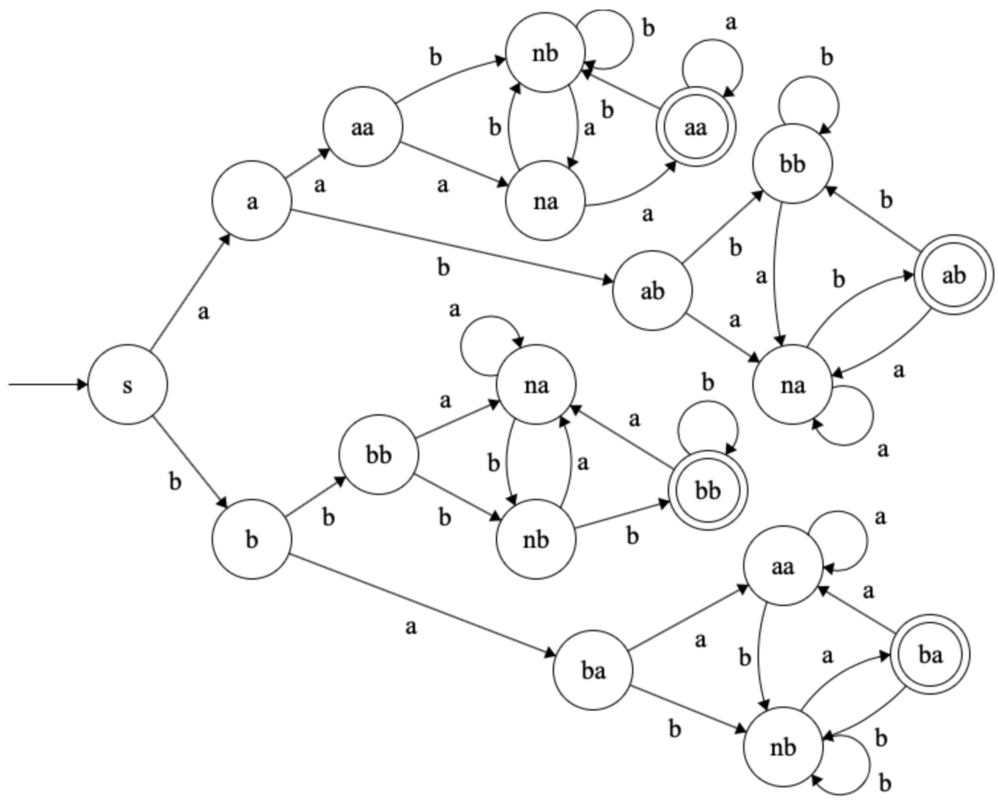
$$\begin{aligned}
LHS &= \hat{\delta}_C((p,q),y'a) \quad \text{where } y' \text{ has length k-1} \\
&= \delta_C(\hat{\delta}_C((p,q),y'),a) \quad \text{by definition(2)} \\
&= \delta_C((\hat{\delta}_A(p,y'), \hat{\delta}_B(q,y')),a) \quad \text{by induction hypothesis} \\
&= (\delta_A(\hat{\delta}_A(p,y'),a), \delta_B(\hat{\delta}_B(q,y'),a)) \quad \text{by definition(3)} \\
&= (\hat{\delta}_A(p,y), (\hat{\delta}_A(p,y)) \quad \text{by definition (2)} \\
&= RHS
\end{aligned}$$

By mathematical induction, we have P(n) is true for all $n \geq 0$

Question 2:



a. The character inside a node denote what characters have already appeared. The node where exactly two characters have appeared are accepted states.



b. The characters in a node represent the last two characters in x(in same order) or last one character if x has length 1. n means it could be either a or b. e.g. nb means it's either ab or bb.

From start state, it takes at least 4 steps to get to an accepted state. This ensures we won't accept string like aaa.

If you input aayaa, you will find you pass two different states denoted as aa(and you could do similar thing for ab, bb, ba as well). The first one denote the first two characters in x is aa. And the other one which is also an accepted state denote that the last two characters are aa.

Question 3:

Given a general DFA $M = (Q, \Sigma, \delta, s, F)$ accepting A, we define $M' = (Q', \Sigma, \delta', s', F')$ where,

$$
\begin{aligned}
Q' &= Q \cup Q \times \Sigma \cup \{d\} \\
\delta'(p, a) &= (p, a) && \text{for all } a \in \Sigma, p \in Q \text{ and } p \neq d \\
\delta'(d, a) &= d && \text{for all } a \in \Sigma \\
\delta'((p, a), b) &= \begin{cases} \delta(p, a) & \text{if } p \in Q, b = a \\ d & \text{if } p \in Q, b \neq a \end{cases} \\
s' &= s \\
F' &= F
\end{aligned}
$$

M' has three types of state. First, it includes all the states of the original machine M and it also has states corresponding to state-character pairs. Other than that, it has a state corresponding to the strings not accepted by machine M'. When M' in a state $q \in Q \times \Sigma$, if it reads a character corresponds to the previous alphabet it will goes back to a state in Q. Otherwise, it will transition into state d, which indicates that the string is not accepted , and stays in state d forever.