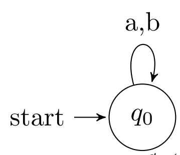
Theory of Computation Assignment no. 2

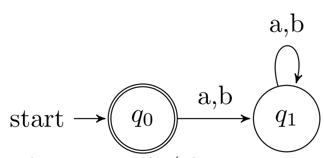
Goktug Saatcioglu

(1) i.



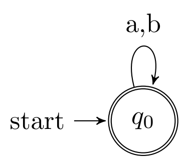
Where $q_0 = \emptyset$ (the empty set).

ii.



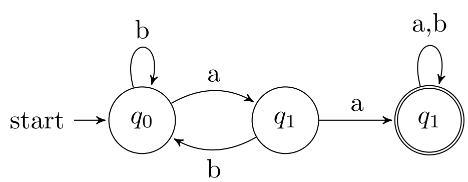
Where $q_0 = \lambda$ (the empty string) and $q_1 \neq \lambda$ (anything but the empty string).

iii.



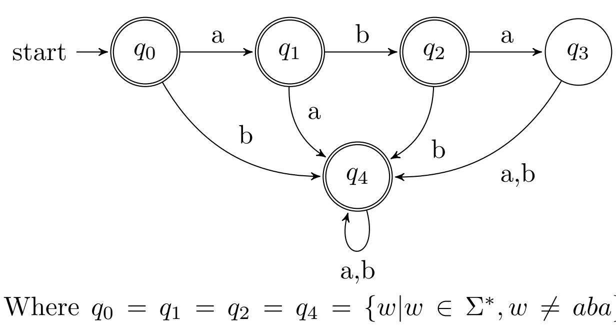
Where $q_0 \in \Sigma^*$ (the set of all strings).

(2) viii.



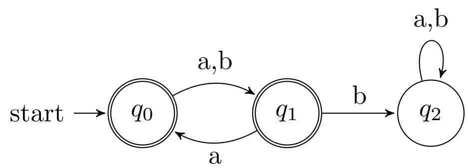
Where $q_0 \in \Sigma^*$ (the set of all strings), $q_1 \in \Sigma_a^*$ (the set of all strings containing a as a substring) and $q_1 \in \Sigma_{aa}^*$ (the set of all strings containing aa as a substring).

xii.



Where $q_0 = q_1 = q_2 = q_4 = \{w | w \in \Sigma^*, w \neq aba\}$ (the set of all strings excepting aba) and $q_3 = aba$ (the aba string).

xiv.



Let the set of all strings with a's in all the even positions and a's or b's in the odd positions be named $VALID$. Any other set that is not $VALID$ is $INVALID$. Thus, $q_0 = q_1 = VALID$ and $q_2 = INVALID$.

(3) Let M_A and M_B be DFA's that recognize the languages A and B respectively, and assume that $M_A = (\Sigma, Q_A, q_A, F_A, \delta_A)$ and $M_B = (\Sigma, Q_B, q_B, F_B, \delta_B)$. Then the formal definition of an automaton $M_{A\Delta B} = (\Sigma, Q_{A\Delta B}, q_{A\Delta B}, F_{A\Delta B}, \delta_{A\Delta B})$, where $A\Delta B = (A \setminus B) \cup (B \setminus A)$, is as follows.

$$\begin{aligned}
Q_{A\Delta B} &\coloneqq Q_A \times Q_B \\
q_{A\Delta B} &\coloneqq (q_A, q_B) \\
F_{A\Delta B} &\coloneqq \{(q,p) | (q \in F_A \land p \notin F_B) \lor (q \notin F_A \land p \in F_B)\} && \text{where } (q,p) \in Q_{A\Delta B} \\
&= (F_A \times (Q_B \setminus F_B)) \cup ((Q_A \setminus F_A) \times F_B) \\
\delta_{A\Delta B} &\coloneqq Q_{A\Delta B} \times \Sigma \to Q_{A\Delta B} \\
&\implies \delta_{A\Delta B}((q,p), w) = (\delta_A(q,w), \delta_B(p,w)) && \text{where } (q,p) \in Q_{A\Delta B},\ w \in \Sigma
\end{aligned}$$

(4) Let $M = (\Sigma, Q, q, F, \delta)$ be a DFA. We prove that for any two words $u, v \in \Sigma^*$ and any state $q \in Q$ that $\delta^*(q, uv) = \delta^*(\delta^*(q,u), v)$, where uv denotes the concatentation of u and v and $|v| = n$ where $n \geq 0$, using induction.
Base case. $n = 0 \implies |v| = 0 \implies v = \lambda$.

$$\begin{aligned}
\forall p \in Q \quad \delta^*(p, \lambda) &= p && (1) \\
\delta^*(q, uv) &= \delta^*(q, u\lambda) \\
&= \delta^*(q, u) \\
&= \delta^*(\delta^*(q,u), \lambda) && \text{using property 1} \\
&\therefore \text{holds true for base case}
\end{aligned}$$

Since the LHS = RHS the base case holds.
Inductive step. For some $n \geq 0$, assume that $\delta^*(q, uv) = \delta^*(\delta^*(q,u), v)$ when $|v| = n$. Now for $k = n$, consider a v' such that $|v'| = k + 1$ and $v' = wa$, where $w \in \Sigma^*$ such that $|v| = l \leq k + 1$ and $a \in \Sigma$. Now we evaluate $\delta^*(q, uv')$.

$$\begin{aligned}
\forall p \in Q \quad \delta^*(p, wa) &= \delta(\delta^*(p, w), a) && \text{where } w \in \Sigma^* \text{ and } a \in \Sigma && (2) \\
\delta^*(q, uv') &= \delta^*(q, uwa) && v' = wa \\
&= \delta(\delta^*(q, uw), a) && \text{using property 2} \\
&= \delta(\delta^*(\delta^*(q,u), w), a) && \text{by the induction hypothesis} \\
&= \delta^*(\delta^*(q,u), wa) && \text{using property 2} \\
&= \delta^*(\delta^*(q,u), v') && wa = v' && (3)
\end{aligned}$$

Conclusion. By the principle of induction, we see that $\delta^*(q, uv) = \delta^*(\delta^*(q,u), v)$ for $|v| = n$ where $n \geq 0$. $\square$