Advanced Topics in Theoretical Computer Science | **Problem Set 1 – Solution**
Due October 17, 2012

Problem 1.1. (15 points) Draw state diagrams for DFAs that accept the following languages.

a.

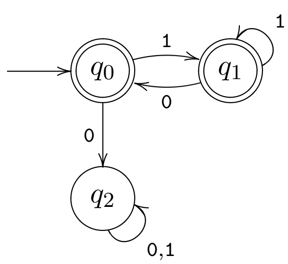
$$\begin{aligned} A &= \{w \in \{\texttt{0},\texttt{1}\}^* \mid \text{in } w \text{ every } \texttt{0} \text{ is immediately preceded by } \texttt{1}\}. \\ B &= \{w \in \{\texttt{1},\texttt{2}\}^* \mid w \text{ contains the subword } \texttt{212}\}. \end{aligned}$$

b.

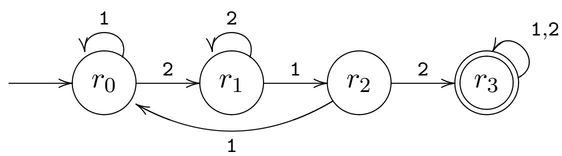
$$\begin{aligned} C &= \{w \in \{\texttt{0},\texttt{1},\texttt{2}\}^* \mid w \text{ contains the subword } \texttt{212}, \\ &\qquad \text{and in } w \text{ every } \texttt{0} \text{ is immediately preceded by } \texttt{1}\}. \\ D &= A \cap B. \end{aligned}$$

Note that $\texttt{21210} \notin D$.

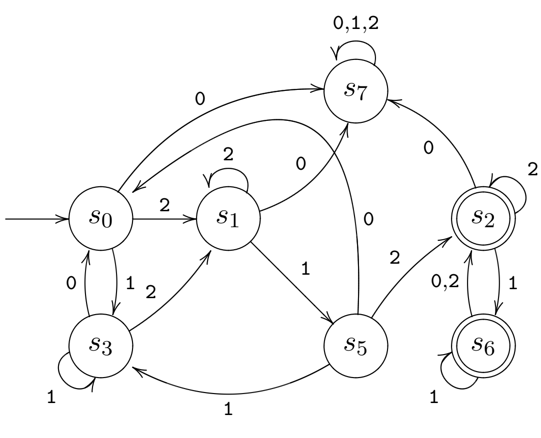
Solution:



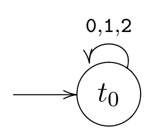
M_A
(3 points)



M_B
(3 points)



M_C
(6 points)



M_D
(3 points)

Problem 1.2. (15 points) For two languages A and B, let

$$A \mid B \;=\; \{a_1a_2a_3\ldots a_{2k} \mid a_1a_3a_5\ldots a_{2k-1} \in A \text{ and } a_2a_4a_6\ldots a_{2k} \in B\}.$$

For example, if `ab` $\in A$ and `cd` $\in B$, then `acbd` $\in A \mid B$.

a. Given a DFA $M_A = (Q_A, \Sigma_A, \delta_A, q_A, F_A)$ that accepts A, and a DFA $M_B = (Q_B, \Sigma_B, \delta_B, q_B, F_B)$ that accepts B, define a generic construction of a finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ that accepts $A \mid B$ (for arbitrary M_A and M_B).

b. Use your construction to draw the state diagram of a DFA that accepts the language $A \mid B$ for the two languages A and B from Problem 1.1(a).

Solution:

a. The DFA $M = (Q, \Sigma, \delta, q_0, F)$ is constructed as follows:

$Q \;=\; Q_A \times Q_B \times \{\mathsf{A}, \mathsf{B}\} \quad \cup \quad \{q_{err}\}$

$\Sigma \;=\; \Sigma_A \cup \Sigma_B$

$\delta : Q \times \Sigma \to Q$ is defined as follows. For each $(q_A, q_B, p) \in Q$ and each $c \in \Sigma$, let

$\delta((q_A, q_B, \mathsf{A}), c) = (\delta_A(q_A, c), q_B, \mathsf{B})$ if $c \in \Sigma_A$

$\delta((q_A, q_B, \mathsf{A}), c) = q_{err}$ if $c \notin \Sigma_A$

$\delta((q_A, q_B, \mathsf{B}), c) = (q_A, \delta_B(q_B, c), \mathsf{A})$ if $c \in \Sigma_B$

$\delta((q_A, q_B, \mathsf{B}), c) = q_{err}$ if $c \notin \Sigma_B$

$\delta(q_{err}, c) = q_{err}$

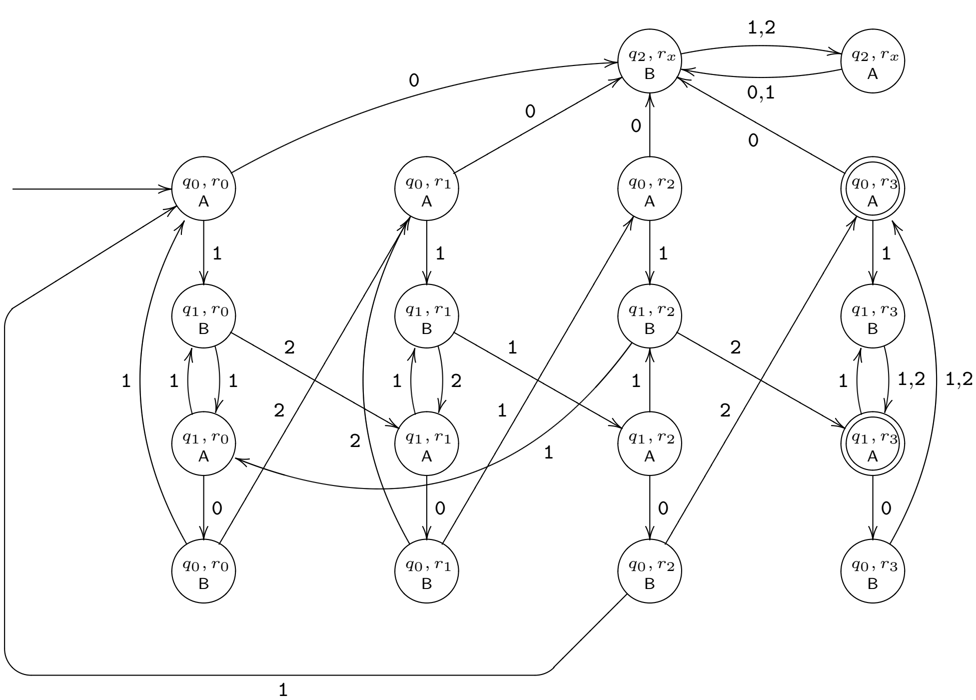
$q_0 \;=\; (q_{0_A}, q_{0_B}, \mathsf{A})$

$F \;=\; \{(q_A, q_B, \mathsf{A}) \mid q_A \in F_A \text{ and } q_B \in F_B\}$

The construction is similar to the product construction, but instead of moving both automata simultaneously they move one after the other. The state remembers (in the third element of the triple) which of the two automata has to move next (and to consume the next input symbol).

(10 points: 4 for the transition relation, 2 for the state set, in particular q_{err}, 2 for the set of final states, 2 for alphabet and initial state)

b.



This state diagram results from applying the construction from (a) to the automata M_A and M_B, and two reductions to avoid clutter in the drawing. First, the state q_{err} and all 2-transitions from A-states and all 0-transitions from B-states are removed. Second, all states that have q_2 as first element are merged to states (q_2, r_x, A) and (q_2, r_x, B). The second reduction is possible because once the first automata is in state q_2, it cannot possibly reach an accepting state.

(5 points)