COMP 335: Introduction to Theoretical Computer Science

Assignment 1

Nathan Grenier

September 20, 2024
Fall 2024

1. [15 Points] For each of the following statements write if the statement is TRUE or FALSE. If the statement is TRUE then provide a proof. If the statement is FALSE then provide a counter-example.

   (a) For every language L we have $L^2 \subseteq L^3$

   Answer: FALSE

   Counter Example:

   $\Sigma = \{a, b\}$

   $L = \{a, ab\}$

   $L^2 = LL = \{aa, aab, aba, abab\}$

   $L^3 = L^2L = \{aaa, aaaba, aaba, aabab, abaa, abaab, ababa, ababab\}$

   $\{aa, aab, aba, abab\} \nsubseteq \{aaa, aaaba, aaba, aabab, abaa, abaab, ababa, ababab\}$

   $\therefore L^2 \nsubseteq L^3$

   (b) For every two languages L_1 and L_2 we have $(L_1 \cup L_2)^* \subseteq (L_1L_2)^*$

   Answer: FALSE

   Counter Example:

   $L_1 = \{x\}$, $L_2 = \{y\}$, $\Sigma = \{x, y\}$

   $L_1 \cup L_2 = \{x, y\}$, $L_1L_2 = \{xy\}$

   $(L_1 \cup L_2)^* = \{\lambda\} \cup \{x, y\} \cup \{xx, xy, yx, yy\} \cup \ldots$

   $(L_1L_2)^* = \{\lambda\} \cup \{xy\} \cup \{xyxy\} \cup \{xyxyxy\} \cup \ldots$

   $\therefore (L_1 \cup L_2)^* \nsubseteq (L_1L_2)^*$

(c) Let L_1 and L_2 be two languages such that $\lambda \in L_1 \cap L_2$. Then it holds that $(L_1 L_2)^* = (L_2 L_1)^*$

Answer: TRUE

Direct Proof:

1. $\subseteq$:

   $\rightarrow$ Let $w \in (L_1 L_2)^*$

   $\rightarrow$ w is of the form: $w = x_1 y_1 x_2 y_2 \ldots x_n y_n$ where $x_i \in L1 \wedge y_i \in L_2$

   $\rightarrow$ Since $\lambda \in L_1 \sup L_2$ then $w = x_1 \, \lambda \, y_1 \, \lambda \, x_2 \, \lambda \, y_2 \ldots x_n \, \lambda \, y_n$

   $\rightarrow$ Format the string $w = (\lambda \, x_1) y_1 (\lambda \, x_2) y_2 \ldots (\lambda \, x_n) y_n$

   $\rightarrow$ Using the associative property of string concatenation $((a \times b) \times c = a \times (b \times c))$, we can rewrite the string as $w = (y_1 x_1) \, \lambda (y_2 x_2) \, \lambda \ldots (y_n x_n) \, \lambda$

   $\rightarrow$ $w = (y'_1 x'_1)(y'_2 x'_2) \ldots (y'_n x'_n)$ where $x'_i \in L_1 \wedge y'_i \in L_2$

   $\therefore$ $w \in (L_2 L_1)^*$

2. $\supseteq$:

   $\rightarrow$ Let $w \in (L_2 L_1)^*$

   $\rightarrow$ w is of the form: $w = y_1 x_1 y_2 x_2 \ldots y_n x_n$ where $x_i \in L1 \wedge y_i \in L_2$

   $\rightarrow$ Since $\lambda \in L_1 \sup L_2$ then $w = y_1 \, \lambda \, x_1 \, \lambda \, y_2 \, \lambda \, x_2 \ldots y_n \, \lambda \, x_n$

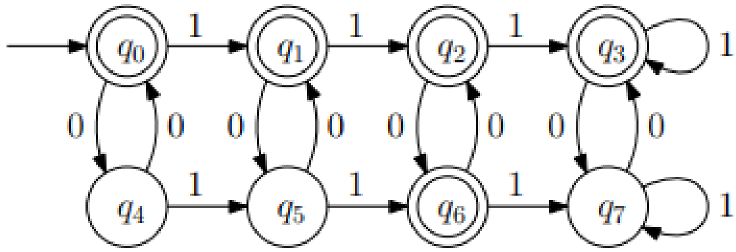
   $\rightarrow$ Format the string $w = (\lambda \, y_1) x_1 (\lambda \, y_2) x_2 \ldots (\lambda \, y_n) x_n$

   $\rightarrow$ Using the associative property of string concatenation $((a \times b) \times c = a \times (b \times c))$, we can rewrite the string as $w = (x_1 y_1) \, \lambda (x_2 y_2) \, \lambda \ldots (x_n y_n) \, \lambda$

   $\rightarrow$ $w = (x'_1 y'_1)(x'_2 y'_2) \ldots (x'_n y'_n)$ where $x'_i \in L_1 \wedge y'_i \in L_2$

   $\therefore$ $w \in (L_1 L_2)^*$

2. [10 Points] The following is a transition diagram for a DFA over the alphabet $\Sigma = \{0, 1\}$. Answer the following questions about this automaton:



(a) What is the start state? What is the set of accept states?

Answer:

Start State = q_0

Accept States = $\{q_0, q_1, q_2, q_3, q_6\}$

(b) What is the sequence of states the DFA goes through on input 101100?

Answer: $(q_0, q_1, q_5, q_6, q_7, q_3, q_7)$

(c) Does the machine accept every string w that contains exactly two 1s? Why or why not?

Answer: $L = \{0^*10^*10^*\}$

Yes. Since the string always contains exactly two 1s, the machine traverses to one of these states $\{q_2, q_6\}$. Since $q_2, q_6 \in F$ and only 0 appear in the string after consuming both 1s, the machine can only transition between these acceptance states.

(d) Does the machine reject every string w that has odd number of 0s? Why or why not?

Answer: $L = \{w \in \Sigma^* : \text{Where } n_0(w) \bmod 2 \neq 0\}$

No. Since there's an odd number of 0s, the any string will always terminate in the following states $\{q_4, q_5, q_6, q_7\}$. However, not all of these states are acceptance states (only $q_6 \in F$). Therefore, depending on how many 1s are present in the string, we may or may not reach an acceptance state.

Counter Example: $w = 011$, Final State = q_6

(e) Describe the language accepted by the machine using the set builder notation.

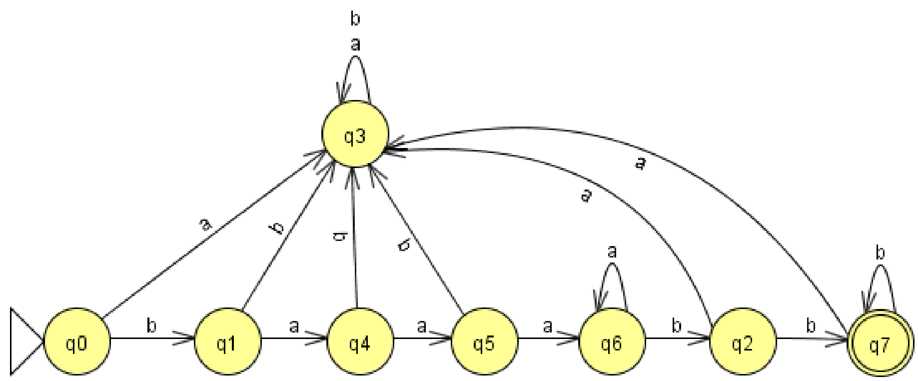
Answer:

$$\begin{aligned} L = &\{w \in \Sigma^* : \text{Where } n_0(w) \,\mathsf{mod}\, 2 = 0\} \vee \\ &\{w \in \Sigma^+ : \text{Where } n_0(w) \,\mathsf{mod}\, 2 \neq 0 \wedge n_1(w) = 2\} \end{aligned} \tag{1}$$

3. [30 Points] For each of the following languages, give a DFA that accepts it.

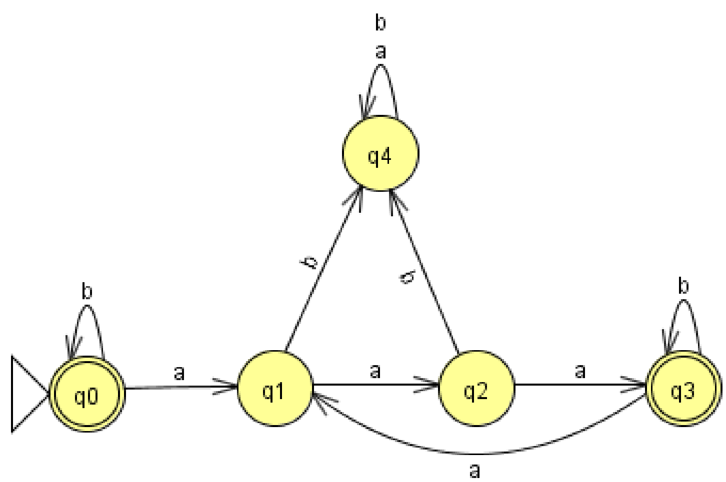
(a) $\{ba^n b^m : n \geq 3, m \geq 2\}$

Answer:



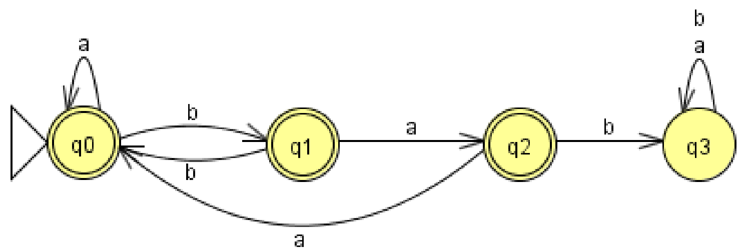
(b) $\{w \in \{a, b\}^* :$ every maximal substring w consisting entirely of symbols a is of length exactly $3\}$

Answer:



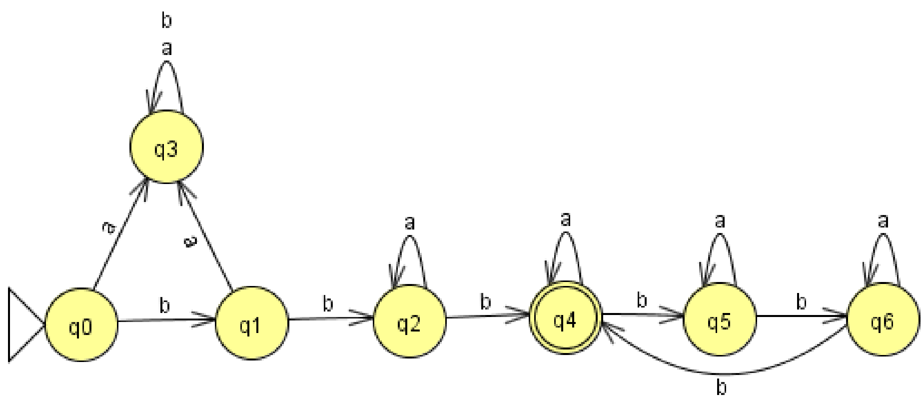
(c) $\{w \in \{a, b\}^* : w$ does not contain bab as a substring$\}$

Answer:



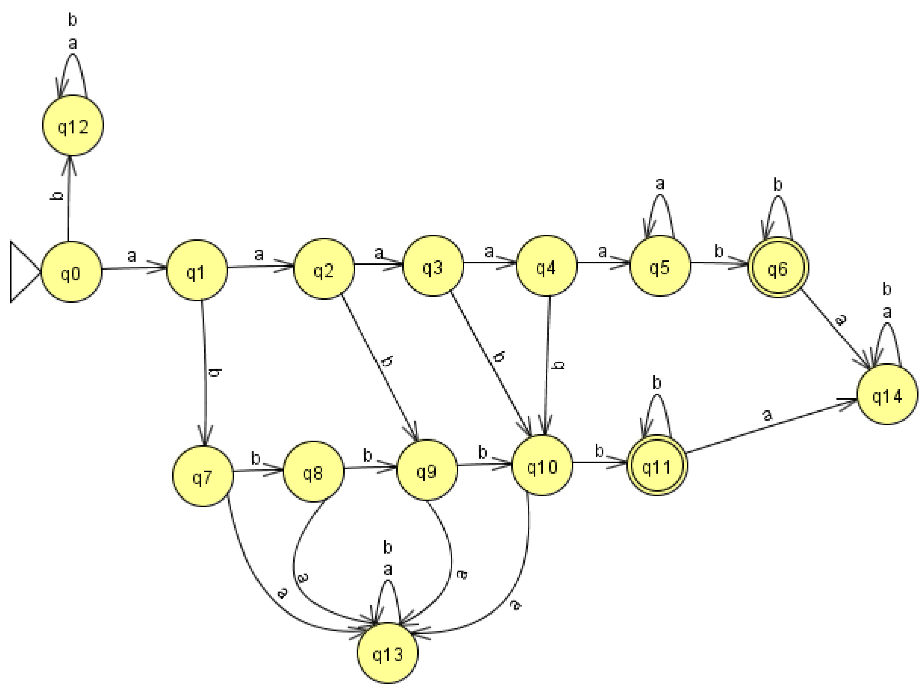
(d) $\{w \in \{a, b\}^* : w$ begins with bb and $n_b(w) \bmod 3 = 0\}$

Answer:



(e) $\{a^m b^n : mn > 4\}$

Answer:



(f) $\{vwv^R : v, w \in \{a, b\}^* \text{ and } |\ v\ |= 2\}$

Answer: