German University in Cairo
Department of Computer Science
Dr. Haythem O. Ismail

Theory of Computation, Winter Term 2013
Assignment3

Discussion: 19.10.13 - 24.10.13

Exercise 3-1

Reading

- Read pages 44 through 47 of the text.

Exercise 3-2

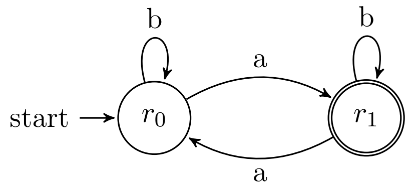
Exercises from Textbook

Sipser (*pp* 83 – 84): Solve exercises 1.4(f, g), 1.5(c, g).

Solution:

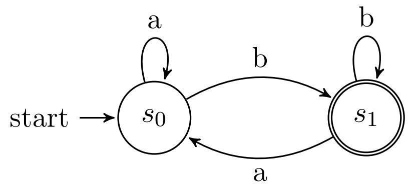
1.4 f) $L = \{w|\ w$ has an odd number of `a`'s and ends with a `b`$\}$
Let M_1 be the DFA that recognizes $L_1 = \{w|\ w$ has an odd number of `a`'s$\}$, with the following representation: $M_1 = (\{r_0, r_1\}, \{a, b\}, \delta_1, r_0, \{r_1\})$, where δ_1 is given by the following table:

	a	b
r_0	r_1	r_0
r_1	r_0	r_1



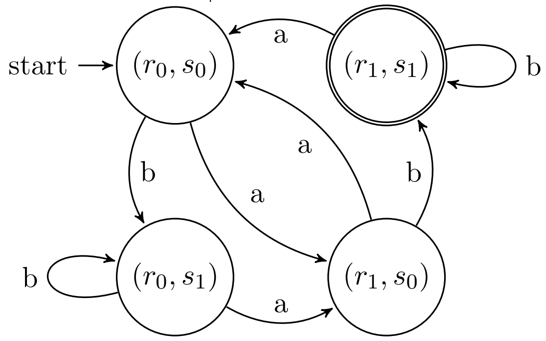
Let M_2 be the DFA that recognizes $L_2 = \{w|\ w$ ends with a `b`$\}$, with the following representation: $M_1 = (\{s_0, s_1\}, \{a, b\}, \delta_2, s_0, \{s_1\})$, where δ_1 is given by the following table:

	a	b
s_0	s_0	s_1
s_1	s_0	s_1



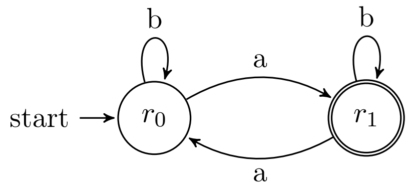
Thus L can be represented as $L = L_1 \cap L_2$. Let M be the DFA that recognizes $L_1 \cap L_2$. Therefore $M = (\{(r_0, s_0), (r_0, s_1), (r_1, s_0), (r_1, s_1)\} \{a, b\}, \delta, (r_0, s_0), \{(r_1, s_1)\})$. δ is given by the following table:

	a	b
(r_0, s_0)	(r_1, s_0)	(r_0, s_1)
(r_0, s_1)	(r_1, s_0)	(r_0, s_1)
(r_1, s_0)	(r_0, s_0)	(r_1, s_1)
(r_1, s_1)	(r_0, s_0)	(r_1, s_1)



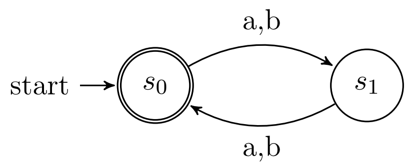
g) $L = \{w|\ w$ has even length and an odd number of a's$\}$
Let M_1 be the DFA that recognizes $L_1 = \{w|\ w$ has an odd number of a's$\}$, with the following representation: $M_1 = (\{r_0, r_1\}, \{a, b\}, \delta_1, r_0, \{r_1\})$, where δ_1 is given by the following table:

	a	b
r_0	r_1	r_0
r_1	r_0	r_1



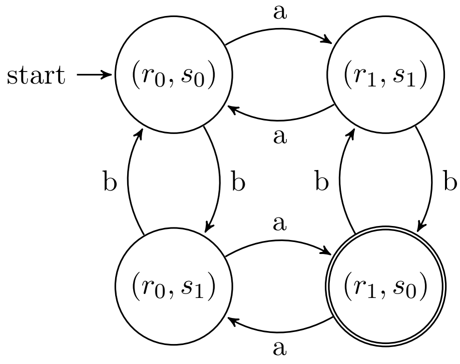
Let M_2 be the DFA that recognizes $L_1 = \{w|\ w$ has even length$\}$, with the following representation: $M_2 = (\{s_0, s_1\}, \{a, b\}, \delta_2, s_0, \{s_0\})$, where δ_1 is given by the following table:

	a	b
s_0	s_1	s_1
s_1	s_0	s_0

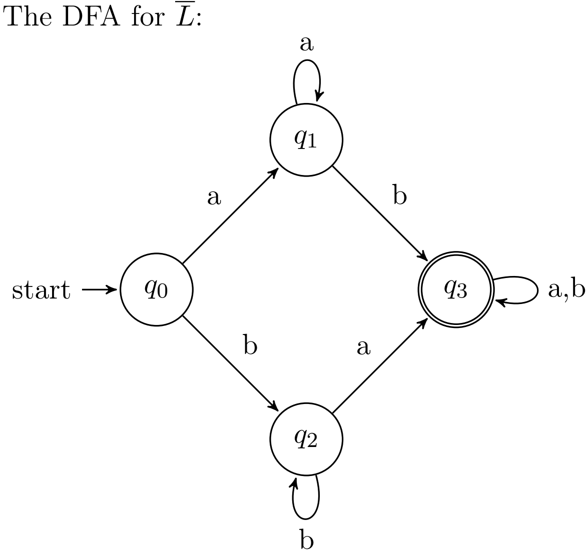


Thus L can be represented as $L = L_1 \cap L_2$. Let M be the DFA that recognizes $L_1 \cap L_2$. Therefore $M = (\{(r_0, s_0), (r_0, s_1), (r_1, s_0), (r_1, s_1)\} \{a, b\}, \delta, (r_0, s_0), \{(r_1, s_0)\})$. δ is given by the following table:

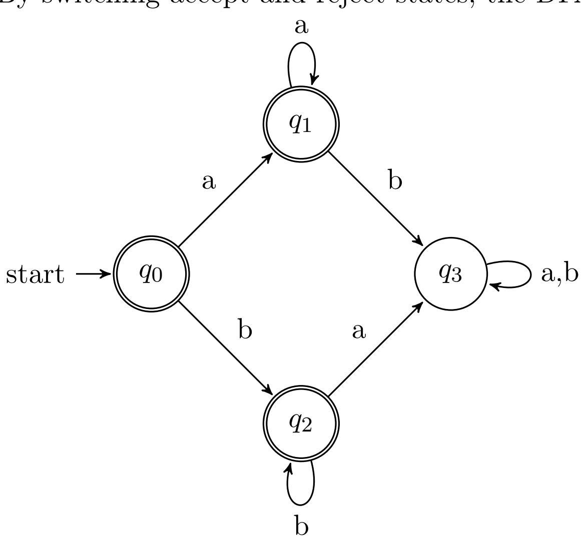
	a	b
(r_0, s_0)	(r_1, s_1)	(r_0, s_1)
(r_0, s_1)	(r_1, s_0)	(r_0, s_0)
(r_1, s_0)	(r_0, s_1)	(r_1, s_1)
(r_1, s_1)	(r_0, s_0)	(r_1, s_0)



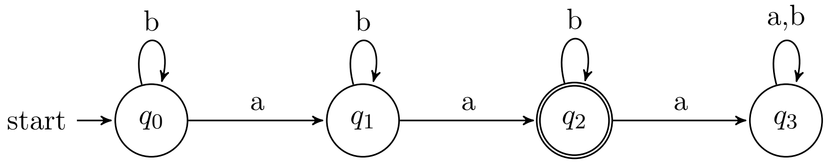
1.5 c) $L = \{w|w$ contains neither the substring `ab` nor `ba`$\}$
$\overline{L} = \{w|w$ contains either the substring `ab` or `ba`$\}$
The DFA for $\overline{L}$:



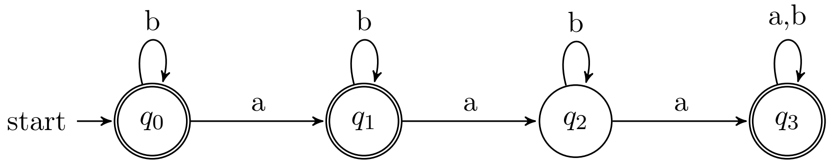
By switching accept and reject states, the DFA for L is as follows:



g) $L = \{w | w \text{ is any string that doesn't contain exactly two } \texttt{as}\}$
$\overline{L} = \{w | w \text{ is any string that contains exactly two } \texttt{as}\}$
The DFA for $\overline{L}$:



By switching accept and reject states, the DFA for L is as follows:



Exercise 3-3

Extra Problem

Prove that any finite language is regular.

Solution:

When saying that a language is finite, this can be said as a language L as n strings. Thus:

$$L = \{w_1, w_2, w_3, \ldots, w_n\} = \bigcup_{i=1}^{n} \{w_i\}$$

Let $L_i = \{w_i\}$, be the language containing only the string w_i. Therefore if there is a DFA M_i that recognizes w_i, then L_i is regular.

Since any string is a finite sequence of symbols elements of an alphabet

$$w_i = s_1 s_2 s_3 \ldots s_m$$

Therefore, the DFA M_i recognizing M_i can be represented as:

$$\begin{aligned}
M_i &= (Q_i, \Sigma_i, \delta_i, q_i, F_i) \\
Q_i &= \{q_j | 1 \leq j \leq m+1\} \cup \{q_r\} \\
\Sigma_i &= \{s_k | 1 \leq k \leq m\} \\
\delta_i(q_j, s_k) &= \begin{cases} q_{j+1} & \text{if } j = k \\ q_r & \text{otherwise} \end{cases} \\
q_i &= q_1 \\
F_i &= \{q_{m+1}\}
\end{aligned}$$

Thus, the Language L_i is regular, as there is the DFA M_i that recognizes it. Since $L = \bigcup_{i=1}^{n} L_i$, and regular languages are closed under union, therefore the language L is also regular. Thus, any finite language is regular.

Exercise 3-4

Programming

Using your favorite programming language, write a method/function/clause that, given two DFA M_1 and $M2$, constructs the DFA M such that $L(M) = L(M_1) \cup L(M_2)$.