1.5 Each of the following languages is the complement of a simpler language. In each part, construct a DFA for the simpler language, then use it to give the state diagram of a DFA for the language given. In all parts $\Sigma = \{a, b\}$.

c. $\{w| w \text{ contains neither the substrings } ab \text{ nor } ba\}$
d. $\{w| w \text{ is any string not in } a^*b^*\}$
e. $\{w| w \text{ is any string not in } (ab^*)^*\}$
f. $\{w| w \text{ is any string not in } a^* \cup b^*\}$
g. $\{w| w \text{ is any string that doesn’t contain exactly two } a\text{'s}\}$
h. $\{w| w \text{ is any string except } a \text{ and } b\}$

1.12 Let $D = \{w| w \text{ contains an even number of } a\text{'s and an odd number of } b\text{'s and does not contain the substring } ab\}$. Give a DFA with five states that recognizes $D$ and a regular expression that generates $D$. (Suggestion: Describe $D$ more simply.)

1.14 a. Show that, if $M$ is a DFA that recognizes language $B$, swapping the accept and nonaccept states in $M$ yields a new DFA that recognizes the complement of $B$. Conclude that the class of regular languages is closed under complement.

b. Show by giving an example that, if $M$ is an NFA that recognizes language $C$, swapping the accept and nonaccept states in $M$ doesn’t necessarily yield a new NFA that recognizes the complement of $C$. Is the class of languages recognized by NFAs closed under complement? Explain your answer.

1.20 For each of the following languages, give two strings that are members and two strings that are not members—a total of four strings for each part. Assume the alphabet $\Sigma = \{a, b\}$ in all parts.

a. $a^*b^*$
b. $a(ba)^*b$
c. $a^* \cup b^*$
d. $(aaa)^*$
e. $\Sigma^*a\Sigma^*b\Sigma^*a\Sigma^*$
ff. $aba \cup bab$
g. $(e \cup a)b$
h. $(a \cup ba \cup bb)\Sigma^*$
1.16 Use the construction given in Theorem 1.39 to convert the following two non-deterministic finite automata to equivalent deterministic finite automata.

![Diagram](a) ![Diagram](b)

1.31 For any string \( w = w_1 w_2 \cdots w_n \), the *reverse* of \( w \), written \( w^R \), is the string \( w \) in reverse order, \( w_n \cdots w_2 w_1 \). For any language \( A \), let \( A^R = \{ w^R | w \in A \} \). Show that if \( A \) is regular, so is \( A^R \).

1.41 For languages \( A \) and \( B \), let the *perfect shuffle* of \( A \) and \( B \) be the language
\[
\{ w | w = a_1 b_1 \cdots a_k b_k, \text{ where } a_1 \cdots a_k \in A \text{ and } b_1 \cdots b_k \in B, \text{ each } a_i, b_i \in \Sigma \}.
\]
Show that the class of regular languages is closed under perfect shuffle.

1.49

a. Let \( B = \{ 1^k y | y \in \{0,1\}^* \text{ and } y \text{ contains at least } k \text{ 1s, for } k \geq 1 \} \).
Show that \( B \) is a regular language.

b. Let \( C = \{ 1^k y | y \in \{0,1\}^* \text{ and } y \text{ contains at most } k \text{ 1s, for } k \geq 1 \} \).
Show that \( C \) isn’t a regular language.