1. Exercise 4.1
   (a) Yes, because $M$ on input 0100 ends in an accept state.
   (b) No, because $M$ on input 011 ends in a non-accept state.
   (c) No, because the input is not in correct form: the second component of the input is missing.
   (d) No, because the input is not in correct form: the first component should be a regular expression but not a DFA.
   (e) No, because $M$ accepts $\lambda$ and hence, $L(M) \neq \emptyset$.
   (f) Yes, because $L(M) = L(M)$.

2. Exercise 4.2
   The problem of testing whether a DFA and a regular expression are equivalent, can be expressed by the following language:
   $$EQ_{\text{DFA-REX}} = \{ < M, r > | M \text{ is a DFA and } r \text{ is a regular expression } \}$$
   and $L(M) = L(r)$.
   We can prove the language $EQ_{\text{DFA-REX}}$ is decidable by constructing a TM $P$ that decides it as follows:
   $P =$ "On Input $< M, r >$:
   
   (a) Convert the regular expression $r$ into a DFA $M_r$ by using the procedure described in theorem 1.28.
   
   (b) Apply the algorithm given in theorem 4.5 to decide whether $< M, M_r > \in EQ_{\text{DFA}}$.
   
   (c) If $< M, M_r > \in EQ_{\text{DFA}}$ the accept, else reject."

3. Exercise 4.3
   proof: Let $M_{\Sigma^*}$ be a DFA that accepts $\Sigma^*$ (this can be easily constructed), then for every DFA $A$,
   $$A \in ALL_{\text{DFA}} \iff < A, M_{\Sigma^*} > \in EQ_{\text{DFA}}$$
   So, to decide whether $A \in ALL_{\text{DFA}}$, we just need to decide whether $< A, M_{\Sigma^*} > \in EQ_{\text{DFA}}$. The latter can be done by applying the proof in theorem 4.5. Thus $ALL_{\text{DFA}}$ is decidable.