Exercise 1.1.

Give a regular expression and an NFA for the language of all words over $\Sigma = \{a, b\}$ ...

(a) ...beginning and ending with $a$.
(b) ...such that the third letter from the right is a $b$.
(c) ...that can be obtained from $babbab$ by deleting letters.
(d) ...with no occurrences of the subword $bba$.
(e) ...with at most one occurrence of the subword $bba$.

Exercise 1.2.

Let $A$, $B$ and $C$ be three languages.
(a) Prove that if $A \subseteq BC$ then $A^* \subseteq (B^* + C^*)^*$. Is the converse true?
(b) Prove that the languages of $((a + ba)^* + b^*)^*$ and $(a + b)^*$ are the same.

Exercise 1.3.

Consider the language $L \subseteq \{a, b\}^*$ given by the regular expression $a*b(ba)^*a$.

(a) Give an NFA that accepts $L$.
(b) Give a DFA that accepts $L$.

Exercise 1.4.

Let $\Sigma = \{a, b\}$ and let $\Sigma^* = (a + b)^*$. Suppose $w = a_1a_2...a_n$ where each $a_i \in \Sigma$. Then the upward closure of a word $w$ is defined as the set

$$\uparrow w = \{u_1u_2...u_na_nu_{n+1} : u_1, u_2, ..., u_{n+1} \in \Sigma^*\}$$

The upward closure of a language $L$ is defined as the set $\uparrow L = \cup_{w \in L} \uparrow w$.

(a) Give an algorithm that takes as input a regular expression $r$ and outputs a regular expression $\uparrow r$ such that $L(\uparrow r) = \uparrow (L(r))$.
(b) Give an algorithm that takes as input an NFA $A$ and outputs an NFA $B$ with exactly the same number of states as $A$ such that $L(B) = \uparrow L(A)$. 