DFAs, NFAs, and Regular Expressions

Many of these proofs are done in the textbook. Try not to look at those parts of the book while working on these problems - it is better practice to derive it yourself (or re-derive it, if you've seen it before) and ask us for a hint if you're stuck.

1. Construct a DFA M with $\Sigma = \{0, 1\}$ such that $L(M) = L(\Sigma^* 0110 \Sigma^*)$. Construct a simpler-looking NFA N which also recognizes the same language.

2. Given a DFA M, prove there is an NFA N such that $L(M) = L(N)$.

3. Given an NFA N, prove there is a DFA M such that $L(N) = L(M)$.

4. Given a regular expression R, prove there is a finite automaton M (your choice whether it's deterministic or not) such that $L(R) = L(M)$.

5. Prove that the set of regular languages is closed under complement, union, and intersection. That is, prove that if L_1 and L_2 are regular, then $\overline{L_1}$, $L_1 \cup L_2$, and $L_1 \cap L_2$ are all also regular. *Recall that a regular language is any language which can be recognized by a DFA. Based on the previous problems (and the other direction for problem 5, which you may assume without proof), it's also any language that can be recognized by an NFA or represented by a regular expression.*

6. What's the point of working with several different models when they're all equivalent to each other anyway? Think of one place (whether a proof or something else in CS/programming) for each of DFAs, NFAs, and regular expressions where that model is the best tool for the job.