Regular Expressions II

Reading: Section 1.3 (all). You do not need to know how to use GNFA\$\text{s} (but you need to understand what they are useful for).

Homework Assignment 8
Due: April 5, in class

Homeworks are due at the beginning of class on the due date. Late homeworks will not be graded for credit, but I will give comments and feedback on them.

Remember to read the descriptions carefully, and think about things that are not mentioned: e.g., if a string is described as containing at most one 1, it may contain no 1s at all; it may also contain any number of 0s!

1. Draw an NFA for each of the following regular expressions. (Each one should be done by applying the right construction to one of the previous ones). For readability, please give the regular expression along with your answer.
   a. 010*1
   b. (010*1)*
   c. (010*1)*1
   d. (010*1)*1(0∪1)*
   e. 1(010*1)*
   f. 1(010*1)*(0∪1)*

2. Do Exercise 1.14 from Sipser.

3. a. Give a regular expression that matches the set of all strings containing a total of exactly two 1s (i.e., any number of 0s are allowed).
   b. Give a regular expression that matches any string containing the substring 010 anywhere within it.

4. Give regular expressions matching the languages of Sipser Exercise 1.4, except for parts f, h, and j.
   Remember to start by working out the small pieces, then combine them properly.
   Example: Build up to part f by writing regular expressions for the following types of strings. Each type builds on the previous ones, so do them in order:
   a. Strings containing exactly two 0s (i.e., plus any amount of 1s anywhere).
   b. Strings containing an even number of 0s. (Use your answer to part a!)
   c. Strings containing exactly two 1s.
   d. Now go back and answer Exercise 1.4f!

5. Writing the DFA for Exercise 1.4f was relatively easy, provided one knows the necessary trick. But writing a regular expression for it is really hard; you’d probably need to use the GNFA construction to do it.
   You are not required to write a regular expression for part f; instead, take a good look at it and explain what it is about this problem that makes it difficult to write a regular expression for it.

6. Optional: Do Exercise 1.16b from Sipser.