1. Give a DFA that accepts all strings over \{0, 1\} that has the property that every segment of five consecutive symbols contains at least two 0’s.

(For example, if the string has length ten, there are altogether six segments of five consecutive symbols.

Also note that, in a formal logical sense, this property is satisfied when the string is of length at most 4. To invalidate the condition there has to exist five consecutive symbols that contains fewer than two 0’s. This is true even if the string is 1111 which has the property that no matter how one might extend it there is no hope to have two 0’s in the first 5 symbols. But that is still not a violation of the condition, for simply there is no counter-example for the string 1111 of length four.)

(Hint: What would you like to keep track of? How many 0’s are there among the last 5 symbols, and the last 4 symbols, .... An alternative way is this: A string is not in this set iff there are some five consecutive symbols having at most one 0.)

2. Give an NFA that accepts all strings over \{a, b\} that end in aba.

Then use the algorithm to convert this NFA to an equivalent DFA.

3. Give an NFA that accepts all strings over \{a, b\} that contain the string abab.

Then use the algorithm to convert this NFA to an equivalent DFA.

4. 1.15.

5. Give NFA’s (with possible ε-moves) accepting the union \(A \cup B\), concatenation \(A \cdot B\), and the Kleene-* operation: \(A^*\) and \(B^*\), of the languages \(A\) of problem 2 and \(B\) of problem 3 in this set.

6. 1.17.

7. Design an NFA which accepts all binary strings which ends in 10011010 exactly.

If you convert this NFA to a DFA, what is the number (or at least approximately the number) of states for your DFA accepting the same set?

(You should not actually carry out this construction. But give an estimate the number of states for your DFA.)