Notes:

- When you are giving a construction, example, etc., provide a justification with your argument. Your solutions to numerical problems must contain the derivation of your answers. In all of your presentations, strive for correctness, completeness, and clarity. When in doubt about the assumptions of problems, the interpretations of wording, etc., consult the instructor.

- You should strive to complete all problems assigned, and a subset of them will be graded.

1. Read the notes above carefully.

2. **You may need to review the prerequisite materials in discrete mathematics to have sufficient working knowledge, and then do the following exercises.**

3. Read [Sip12] Chapter 0; you may need to review the prerequisite materials in discrete mathematics to have sufficient working knowledge, and then do the following exercises.

4. Do [Sip12] Chapter 0, problem 0.13.

5. An alphabet is a non-empty finite set of symbols, and a string over the alphabet is a finite sequence of symbols of the alphabet. Some example strings over the binary alphabet \( \{0, 1\} \) are: 1011 (for the sequence \( (1, 0, 1, 1) \)), 10 (for the sequence \( (1, 0) \)), \( \epsilon \) (denoting the empty sequence).

For strings \( x \) and \( y \) over an alphabet, we denote by \( |x| \) the length of the sequence \( x \), and by \( xy \) the concatenation of the two sequences \( x \) and \( y \) in that order.

For each integer \( n \geq 0 \), we define the strings \( x_n \) and \( y_n \) over the alphabet \( \{0, 1\} \) as follows: \( x_0 = 0 \) and \( y_0 = 1 \), and for \( n \geq 1 \), \( x_n = x_{n-1}y_{n-1} \) and \( y_n = y_{n-1}x_{n-1} \). Prove the following statements using mathematical induction:

   (a) For every \( n \geq 0 \), \( |x_n| = |y_n| \).

   (b) For every \( n \geq 0 \), \( x_n \) and \( y_n \) differ in every position.

   (c) For every \( n \geq 0 \), \( x_{2n} \) and \( y_{2n} \) are palindromes. (A string \( x \) is a palindrome if the reversal sequence of \( x \) is identical to the sequence \( x \).)

   (d) For every \( n \geq 0 \), \( x_n \) contains neither the substring 000 nor the substring 111. (A string \( x \) is a substring of a string \( y \) if \( x \) is simply a contiguous subsequence of \( y \).)

6. Give some examples of strings in, and not in, the following languages over the alphabet \( \Sigma = \{a, b\} \). Explain your answers.

   (a) \( \{w \in \Sigma^* \mid \text{for some } u \in \Sigma^2, w = uu^r u\} \).

   (b) \( \{w \in \Sigma^* \mid w = www\} \).

   (c) \( \{w \in \Sigma^* \mid \text{for some strings } u \text{ and } v \text{ over } \Sigma, uvw = wvu\} \).

   (d) \( \{w \in \Sigma^* \mid \text{for some string } u \text{ over } \Sigma, uvw = uu\} \).

7. Let \( \Sigma \) be the alphabet \( \{0, 1\} \). Denote by \( L \) the language \( \{u \in \Sigma^* \mid u = vv \text{ for some string } v \in \Sigma^*\} \). Prove or disprove that the language \( L \) can be expressed as the concatenation of two “non-trivial” languages \( L_1 \) and \( L_2 \) over \( \Sigma \): \( L_1 \neq \{\epsilon\} \) and \( L_2 \neq \{\epsilon\} \) and \( L = L_1L_2 \).

8. For each of the following languages, construct a deterministic finite automaton that accepts the language. You need to give brief and precise interpretations for the states of the machine.

   (a) \( L_1 = \{x \in \{0, 1\}^* \mid x \text{ contains three consecutive 1s}\} \).

   (b) \( L_2 = \{x \in \{0, 1\}^* \mid x \text{ does not end with 11}\} \).
9. Let $L \subseteq \{0, 1\}^*$ be the language of all strings such that there exist two 0s separated by a number of positions that is a non-zero multiple of 5. For example, the string 1001110 is not in $L$, but the string 1011011011111110 is in $L$. Intuitively, any deterministic finite automaton accepting $L$ must “remember” 5 positions in order to determine the membership of the string — this is a general idea but not a proof.

Prove that every deterministic finite automaton accepting $L$ must have at least $2^5$ states.