Notes:

- Read Course Information: Section 7 (Miscellaneous) and Section 9 (Academic Dishonesty or Misconduct).
- 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. Give the 5-tuple definition and state-transition diagram for a finite automaton (deterministic, nondeterministic, or nondeterministic finite automaton with $\epsilon$-transitions) accepting each of the following languages. Briefly and precisely give the interpretations of the states and transitions in your construction.
   
   (a) The set of all strings in $\{0, 1\}^*$ such that, at some place in the string, there are two 0s separated by an even number of symbols.
   
   (b) The set of all strings in $\{a, b, c\}^*$ such that exactly one of the two symbols $a$ or $b$ appears at least three times in all.
   
   (c) The set $\text{half}(L)$ of all first halves of strings in a regular language $L$ over an alphabet $\Sigma$, that is,

   $$\text{half}(L) = \{ x \in \Sigma^* \mid \exists y \ (xy \in L \text{ and } |x| = |y|) \}.$$

3. The notion of acceptance for the model of nondeterministic finite automata (NFAs) is based on the existence of a computation (an execution) of the machine on the input that leads to an accepting state upon the completion of processing the input. We now define a new class of NFAs, called $\forall$-NFAs, in which an input string is accepted if all possible computations (executions) lead to an accepting state upon the completion of processing the input.

   (a) Show that every regular language is accepted by an $\forall$-NFA.
   
   (b) Show that the language accepted by an $\forall$-NFA is regular.

4. Do [Sip12] Chapter 1, exercises 1.16 (b) and 1.21 (a). Show all the (important) intermediate work.

5. Do [Sip12] Chapter 1, exercise 1.18 (h) and (i) (in exercise 1.6). Justify your solutions by properly structuring and annotating your regular expressions.

6. ... More problems may be given in later version.