DISCLAIMER: These notes are not necessarily an accurate representation of what I said during the class. They are mostly what I intend to say, and have not been carefully edited.

- Comments from last time: ???

1. Test 1: next class, Wednesday Sep 19, in class. List of topics (exercise / problem numbers are for the Third Edition of the text book):

   1. Definition of languages:
      (a) Set theoretic definitions / notations (Homework 1 Problem 1, Quiz 1).
      (b) A language is a set of words.
      (c) A word is a string of characters from the alphabet.

   2. Automata:
      (a) State, alphabet, transition, starting state, and accepting states. (Exercise 1.1, 1.2, 1.3)
      (b) DFA: moves to a unique state $y = \delta(x, a)$ when at state $x$, and input is $a$.
      (c) NFA: moves to a set of possibly empty states, plus $\epsilon$-transitions.
      (d) Simulating DFAs/NFAs: follow the arrows, maintain (set) of current states (Homework 1 Problems 2, Homework 2 Problem 3a).
      (e) Equivalent DFA of an NFA (Homework 2 Problem 3b).
      (f) Constructions of simple DFAs/NFAs (Homework 1 Problems 3 and 4, Exercise 1.12 first half, 1.13, Problem 1.36, 1.37, 1.40, 1.60).

   3. Regular ...
      (a) Regular Languages: languages accepted by some DFA.
      (b) Regular operations: union / concatenation / $*$ on languages.
      (c) Regular expression: start with ‘base languages’ $a \in \Sigma$, $\epsilon$, or $\emptyset$, then recursively combine them using regular operations.

   4. Manipulations of automata / Regular Languages:
      (a) DFA for the complement of a regular language: set all accepting states to non-accepting, and all non-accepting states to accepting. (Exercise 1.5, 1.14)
(b) Explicit construction of a DFA accepting the union of two DFA (Homework 1 Problem 4, Homework 2 Problem 1, Exercise 1.4).

(c) Constructing NFAs for the results of applying each of the regular operations to regular languages (Homework 2 Problem 2, Exercise 1.8, 1.9, 1.10).

(d) Construction of DFAs/NFAs for languages corresponding to regular expressions (Exercise 1.17, 1.19, 1.28).

5. **Not Covered:**

   (a) Construction of regular expression from specification of language.

   (b) Construction of regular expression from NFA.

   (c) Conversion of NFAs to one with no transitions entering starting state, or only one accepting state.

   (d) Generalized NFA (GNFA): transitions can be regular expressions.

   (e) Non-regular languages (anything starting in Chapter 1.4. of textbook and after).