COMS 3261: Computer Science Theory

Problem Set 1, due Monday, 9/30/13, at the beginning of the class

Please follow the Homework Guidelines.

Try to make your answers as precise, succinct, and clear as you can.

Part A: [30 points] Do the problems posted at Gradiance.

Part B: Turn in the following problems.

Problem 1. [20 points] Consider the DFA specified by the transition table in Exercise 2.2.11 of the book (page 55).
a. Draw the state transition diagram of the automaton. Make sure to specify in your diagram the start state and the accepting states.
b. Give the sequence of states corresponding to the computation of the automaton on the input string 01101001
c. List 3 input strings that are accepted by the automaton and 3 input strings that are not accepted.
d. Describe the language accepted by the DFA. Justify your answer informally. You do not need to give a formal proof.

Problem 2. [15 points]
Design deterministic finite automata for the following languages. You can give the DFA by their transition diagrams. You do not need to show that they are correct.
a. {101, 11, 010, 0110}
b. The set of strings over the alphabet $\Sigma=\{a,b,\ldots,z\}$ that contain at least one *m* between any two *a*'s in the string; the language contains in particular all strings that contain only one *a* or no *a*'s. For example *abc, john, mama, american* are in the language, but *papa*, *panamerican* are not.
c. The set of strings over the alphabet $\Sigma=\{a,b,..,z\}$ that contain the string *rare* as a substring, i.e. $L=\{\pi \mid \pi = \varphi rare\psi$ for some strings $\varphi,\psi \in \Sigma^* \}$.

Problem 3. [20 points] Consider the NFA in Exercise 2.3.2 in the textbook (page 66).
a. [5 points] Give the tree of all computations of the NFA for the input 0110.
b. [3 points] What is $\hat{\delta}(p,0110)$? Is 0110 accepted?
c. [12 points] Convert the NFA to a DFA using the subset construction. Give the DFA by its transition table, and include only the states that are reachable from the initial state.

Problem 4. [15 points] Design nondeterministic finite automata for the following languages. Try to take advantage of nondeterminism to make your automaton as simple as possible, and use as few states and transitions as you can. You may also use ε-transitions if it helps you.

a. $\{\varepsilon\}$

b. The set of strings over the alphabet $\Sigma=\{a,b,..,z\}$ that do not contain all the letters of the alphabet.

c. The set of nonempty strings over the alphabet $\Sigma=\{0,1,\ldots,9\}$ such that the last digit of the string has not appeared before; for example 2013 is in the language, but 2012 is not.