Problem Set 3
CS 373: Theory of Computation

Assigned: September 12, 2013 Due on: September 19, 2013

Instructions: This homework has 3 problems that can be solved in groups of size at most 3. Please follow the homework guidelines given on the class website. Solutions not following these guidelines will not be graded.

Recommended Reading: Lectures 5, and 6.

Problem 1. [Category: Design+Proof] For a string $w \in \Sigma^*$, let w^R denote the reverse of w, i.e., if $w = w_1 w_2 \cdots w_n$, where $w_i \in \Sigma$ then $w^R = w_n w_{n-1} \cdots w_1$. For a language L, let $L^R = \{w^R \mid w \in L\}$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA.

1. Design a *DFA* M^R that recognizes $\mathbf{L}(M)^R$, i.e., $\mathbf{L}(M^R) = (\mathbf{L}(M))^R$. **[5 points]**
2. Prove that your DFA M^R in the previous part is correct. **[5 points]**

Problem 2. [Category: Comprehension+Design] An ***all***-NFA M is a 5 tuple $(Q, \Sigma, \delta, q_0, F)$ like an NFA, where Q is a finite set of states, Σ is the input alphabet, $\delta : Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q$ is the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. The only difference between an ***all***-NFA and an NFA is that M accepts $u \in \Sigma^*$ iff every possible state that M could be in after reading u is in F (and at least one state is in F, i.e., all threads cannot die).

1. Taking $q_1 \xrightarrow{w}_M q_2$ to be the same as definition 4 in lecture 4, define formally when an ***all***-NFA M accepts u, and the language recognized by M (definitions similar to definitions 5 and 6 in lecture 4). **[4 points]**
2. Give a formal definition of a DFA $\text{dfa}(M)$ such that $\mathbf{L}(\text{dfa}(M)) = \mathbf{L}(M)$. You need not prove your construction to be correct. **[6 points]**

Problem 3. [Category: Comprehension+Design]

1. Describe the language of the following regular expressions. A clear, crisp one-level interpretable English description is acceptable, like "This is the set of all binary strings with at least three 0s and at most hundred 1s", or like "$\{0^n(10)^m \mid n$ and m are integers$\}$". A vague, recursive or multi-level-interpretable description is not, like "This is a set of binary strings that starts and ends in 1, and the rest of the string starts and ends in 0, and the remainder of the string is a smaller string of the same form!" or "This is a set of strings like 010, 00100, 0001000, and so on!". You need not prove the correctness of your answer.
   (a) $(0^* \cup 0 \cup 1^*)^*$ **[1 points]**
   (b) $0(10)^*1$ **[2 points]**
   (c) $1^*(0 \cup 111^*)^*1^*$ **[2 points]**

2. Give regular expressions that accurately describe the following languages. You need not prove the correctness of your answer.

   (a) All binary strings with no more than three 0s. **[1 points]**

   (b) All binary strings such that in every prefix, the number of 0s and 1s differ by at most 1. **[2 points]**

   (c) All binary strings with exactly one occurrence of the substring 000. **[2 points]**