6.045J/18.400J: Automata, Computability and Complexity Prof. Ron Rivest

Handout 3: Quiz practice problems

Due: 27 February 2002

These are the problems we would have handed out for homework this week, had we assigned homework. All of these problems are optional.

Problem 1: Problem 1.16 (page 86) parts a and b

Problem 2: For the following regular expressions, produce an NFA that recognizes the same language:

1. The expression of problem 1.15, part a
2. The expression of problem 1.15, part b
3. The expression of problem 1.15, part e

Problem 3: Consider the following regular expression operations from the unix `grep` command. For each, prove that regular expressions as defined in class, with the new operation, still define regular languages:

1. `?`: If R is a regular expression, then $R?$ is zero or one instances of R
2. `(n,m)`: If R is a regular expression, then $R(\mathtt{n},\mathtt{m})$ is between n and m (inclusive) instances of R,
3. $\{\}$: If R is a regular expression, then $\{R\}$ is a string containing a contiguous substring that matches R. For example, $\{\mathtt{ac}\}$ contains `bbbacb` but not `bbabbcbb`. (Note: this is not from `grep`, since this is how `grep` usually regards regular expressions.)

Problem 4: Consider the informal definition of a *finite state transducer* from Problem 1.19 (page 87). Formally, a finite state transducer is a five-tuple

$$(Q, \Sigma, \Gamma, \delta, q_0)$$

where

- Q is a finite set of states,
- Σ is a finite set, called the *input alphabet*,
- Γ is a finite set, called the *output alphabet*,
- $\delta : Q \times \Sigma \to Q \times \Gamma$ is a function from states and the input alphabet to states and the output alphabet. Informally, this function takes the current state and the next character of the input, and gives you the next state and the next character of the output.
- $q_0 \in Q$ is the start state.

Note that there is no concept of "accepting" states. Finite state transducers map input strings to output strings. For this problem, however, we will make one change. In particular, we will allow transitions to produce no output. That is, the transition function is now

$$\delta : Q \times \Sigma \to Q \times (\Gamma \cup \{\epsilon\})$$

So, let T be a finite state transducer. Let $L(T)$ be the set of strings that can be produced by T. That is, run T on all possible input and let $L(T)$ be the outputs it could produce. Prove that for any finite state transducer T, $L(T)$ is regular.