Problem Set 3
CS 373: Theory of Computation

Assigned: January 31, 2013 Due on: February 7, 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; submitions not following these guidelines will not be graded.

Recommended Reading: Lectures 5 and 6.

Problem 1. [Category: Comprehension+Design+Proof] 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). **[2 points]**
2. Give a formal definition of a DFA $\text{dfa}(M)$ such that $\mathbf{L}(\text{dfa}(M)) = \mathbf{L}(M)$. Prove that your construction is correct. **[3+5 points]**

Problem 2. [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 "$\{0^n(10)^m \,|\, n \text{ and } m \text{ are integers}\}$". 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) $0^*(10^*)^*$ **[1 points]**
   (b) $0(10)^*1$ **[2 points]**
   (c) $c^*(a \cup (bc^*))^*$ **[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 that have two or three occurrences of 1 such that the first and the second occurrence (of 1) are not consecutive. **[2 points]**
   (c) All binary strings with exactly one occurrence of the substring 000. **[2 points]**

Problem 3. [Category: Proof] Let r and s be regular expressions. Consider the equation $X = rX \cup s$. A regular expression t is a solution to this equation if $t = rt \cup s$ (i.e., $\mathbf{L}(t) = \mathbf{L}(rt \cup s)$).

1. Assuming that $\epsilon \notin \mathbf{L}(r)$, prove that if t is a solution to the above equation then $\mathbf{L}(t) = \mathbf{L}(r^*s)$. **[7 points]**

   Hint: To show that $\mathbf{L}(t) \subseteq \mathbf{L}(r^*s)$, first show that $\mathbf{L}(t) \subseteq \mathbf{L}(r^n t \cup r^{n-1}s \cup \ldots \cup s)$, for all $n \geq 0$. What can you say about strings $w \in L$ such that $|w| < n$?

2. If $\epsilon \in \mathbf{L}(r)$ then the equation need not have a unique solution. Assuming $\epsilon \in \mathbf{L}(r)$, give a solution for $X = rX \cup s$ that does not depend on r or s. **[3 points]**

(Your homework ends at this point. What follows is an application of the result you have proved in this problem for your reading pleasure.)

Application of Problem 3: The above result can be used to obtain another algorithm for converting DFAs to regular expression; lecture 7 describes a different algorithm. Consider the DFA M shown below. Let variable X_i denote the language consisting of strings that reach the accept state when starting from the

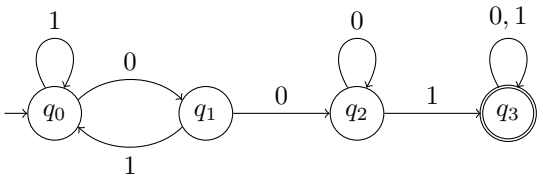

Figure 1: DFA M

state q_i. Now, based on the transitions of M, we write down constraints on the variables X_i. For example, X_0 consists of strings that begin with 0 and then are followed by a string in that is accepted from q_1 (in other words, belongs to X_1) or begin with a 1 and are followed by a string that is accepted from q_0 (in other words belongs to X_0). Thus, we can write the equation

$$X_0 = 0X_1 \cup 1X_0$$

By observing the transitions out of every other state, we will get the following additional equations.

$$X_1 = 0X_2 \cup 1X_0$$
$$X_2 = 0X_2 \cup 1X_3$$
$$X_3 = (0 \cup 1)X_3 \cup \epsilon$$

Among the above equations, the only thing to note is the equation for the final state X_3: $(0 \cup 1)X_3$ term on the R.H.S is obtained from the transitions out of q_3, and ϵ is added because we know that ϵ belongs the language X_3 because q_3 is a final state.

Now if we can find regular expressions for X_0, X_1, X_2 and X_3 that simulatenously satisfy all the above equations, then we would have identified the strings that are accepted from each state. The regular expression equivalent to M would then be the regular expression associated with the variable of the initial state, which in this case is X_0.

How do we solve such equation systems? By "Gaussian elimination" and the result in Problem 3 that you have just proved. Let us look at the last equation: it is exactly of the form in Problem 3, if we take

$r = 0 \cup 1$ and $s = \epsilon$. Thus, from the result in Problem 3, the solution for X_3 is the regular expression $(0 \cup 1)^*\epsilon = (0 \cup 1)^*$. Now we substitute X_3 in the equation for X_2 and solve it using the result in Problem 3 to get the expression $0^*(1(0 \cup 1)^*) = 0^*1(0 \cup 1)^*$. Then we substitute the solution for X_2 in the equation for X_1, and we get $X_1 = 0(0^*1(0 \cup 1)^*) \cup 1X_0$ or in other words, $X_1 = \emptyset X_1 \cup (00^*1(0 \cup 1)^* \cup 1X_0)$. So taking $r = \emptyset$ and $s = (00^*1(0 \cup 1)^* \cup 1X_0)$, we find that this in the form in Problem 3, and its solution is $\emptyset^*(00^*1(0 \cup 1)^* \cup 1X_0) = (00^*1(0 \cup 1)^* \cup 1X_0)$. Note, we have not found a solution for X_1 yet, because it depends on the solution for X_0. But, we can use what we have found to eliminate X_1 from the equation for X_0. Substituting for X_1 in the equation for X_0 and rearranging terms (using distributivity laws), we get

$$\begin{aligned} X_0 &= 0(00^*1(0 \cup 1)^* \cup 1X_0) \cup 1X_0 \\ &= 000^*1(0 \cup 1)^* \cup 01X_0 \cup 1X_0 \\ &= (01 \cup 1)X_0 \cup 000^*1(0 \cup 1)^* \end{aligned}$$

By a final application of Problem 3, we get the regular expression for X_0 to be $(01 \cup 1)^*(000^*1(0 \cup 1)^*)$. At this point, we could get the solution for X_1, but we don't need to because the regular expression for X_0 is our desired regular expression as q_0 is the initial state.

In general for an n-state DFA, we would get an equation system that looks like

$$\begin{aligned} X_1 &= \alpha_{11}X_1 \cup \alpha_{12}X_2 \cup \cdots \cup \alpha_{1n}X_n \cup c_1 \\ X_2 &= \alpha_{21}X_1 \cup \alpha_{22}X_2 \cup \cdots \cup \alpha_{2n}X_n \cup c_2 \\ &\vdots \quad \vdots \quad \vdots \\ X_n &= \alpha_{n1}X_1 \cup \alpha_{n2}X_2 \cup \cdots \cup \alpha_{nn}X_n \cup c_n \end{aligned}$$

Treat the last equation as an equation in only one variable X_n: $X_n = \alpha_{nn}X_n \cup (\alpha_{n1}X_1 \cup \alpha_{n2}X_2 \cup \cdots \cup \alpha_{nn-1}X_{n-1} \cup c_n)$. Thus, taking $r = \alpha_{nn}$ and $s = (\alpha_{n1}X_1 \cup \alpha_{n2}X_2 \cup \cdots \cup \alpha_{nn-1}X_{n-1} \cup c_n)$, we can use Problem 3 to obtain a solution for X_n in terms of $X_1, \ldots X_{n-1}$. We substitute this solution back in all the equations, and we would have eliminated X_n from the equation system. We repeat this process until we get an equation for the initial state variable only in terms of the initial state variable, and solve this final equation to get the regular expression!