CS 301 Languages and Automata, UIC
Fall 2012, Assignment 3
Due: Friday, September 28, 2012 at start of class

Unless otherwise noted, the alphabet for all questions below is assumed to be $\Sigma = \{0, 1\}$.

1. This question develops your ability to devise regular expressions, given an explicit definition of a language. For each of the following languages, prove they are regular by giving a regular expression which describes them. Justify your answers.

   (a) $L = \{x \mid x \text{ begins with a } 0 \text{ and ends with a } 1\}$.

   (b) $L = \{x \mid x \text{ contains at least four } 0\text{'s}\}$.

   (c) $L = \{1, 11, \epsilon\}$.

   (d) $L = \{x \mid \text{the length of } x \text{ is at most } 3\}$.

   (e) $L = \{x \mid x \text{ doesn't contain the substring } 110\}$.

   (f) $L = \{x \mid |x| > 0, \text{ i.e. } x \text{ is non-empty}\}$.

2. This question tests your understanding of how to translate a regular expression into a finite automaton. Using the construction of Lemma 1.55, construct NFAs recognizing the languages described by the following regular expressions.

   (a) $R = \emptyset^*$.

   (b) $R = (0 \cup 1)^* 111 (0 \cup 1)^*$.

3. This question tests your understanding of how to translate a finite automaton into a regular expression. Consider DFA $M = (Q, \Sigma, \delta, q, F)$ such that $Q = \{q_1, q_2, q_3\}$, $q = q_1$, $F = \{q_1, q_3\}$, and δ is given by:

δ	0	1
q_1	q_2	q_2
q_2	q_2	q_3
q_3	q_1	q_2

   Draw the state diagram for M, and then apply the construction of Lemma 1.60 to obtain a regular expression describing $L(M)$.

4. This question allows you to practice proving a language is non-regular via the Pumping Lemma. Using the Pumping Lemma (Theorem 1.70), give formal proofs that the following languages are *not* regular:

   (a) $L = \left\{www \mid w \in \{0, 1\}^*\right\}$.

   (b) $L = \{1^n 0^m 1^n \mid m, n \geq 0\}$.

   (c) $L = \left\{x \mid x \in \{0, 1\}^* \text{ is not a palindrome}\right\}$. Recall a palindrome is a string that looks the same forwards and backwards. Examples of palindromes are "madam" and "racecar".

5. This question further tests your understanding of the subtleties of the Pumping Lemma:

(a) Let $B_1 = \left\{0^k 1 x 0^k \mid k \geq 1 \text{ and } x \in \Sigma^*\right\}$. Give a formal proof that B_1 is not a regular language.

(b) Let $B_2 = \left\{0^k x 0^k \mid k \geq 1 \text{ and } x \in \Sigma^*\right\}$. Show that B_2 is a regular language.

(c) We just proved in part (b) above that B_2 is regular. Consider now the following argument, which claims to prove that B_2 is, in fact, not regular.

Assume B_2 is regular, and let p be the pumping length. Consider string $s = 0^p 1 0^p \in B_2$, and decompose it as $s = xyz$ with $x = \epsilon$, $y = 0^p$, $z = 10^p$. Then, pumping s down by setting $i = 0$ yields string $s' = xy^i z = xy^0 z = 10^p \notin B_2$. Hence, by the Pumping Lemma, we have a contradiction. We conclude that B_2 is not regular.

i. What is wrong with this proof?

ii. Why does pumping up not work, i.e. why does setting $i > 1$ in the argument above yield a string $s' \in B_2$?

6. In showing how to convert any NFA to a regular expression, we introduced the concept of a GNFA. Crucially, our construction used the fact that any GNFA can be converted to an equivalent GNFA consisting of precisely two states. Can we similarly shrink the number of states in a DFA?

(a) Prove that for every $k > 1$, there exists some language $L_k \subseteq \{0,1\}^*$ such that L_k is recognized by some DFA with k states, but there is no DFA with only $k-1$ states recognizing L_k.

(b) Which question from Assignment 2 immediately demonstrates that there does not exist a procedure which, given any DFA as input, outputs an equivalent DFA consisting of just two states?