CSE 547T Automata Theory Spring Semester 2017

Homework 2: More on Finite Automata

Assigned: 2/13/2017 *Due Date: 3/1/2017*

This homework must be completed and submitted electronically. Formatting standards, submission procedures, and (optional) document templates for homeworks may be found at

`http://classes.engineering.wustl.edu/cse547/ehomework/ehomework-guide.html`

Advice on how to compose homeworks electronically, with links to relevant documentation for several different composition tools, may be found at

`http://classes.engineering.wustl.edu/cse547/ehomework/composing-tips.html`

Please remember to

- **create a separate PDF file (typeset or scanned) for each problem;**
- **include a header with your name, WUSTL key, and the homework number at the top of each page of each solution;**
- **include any figures (typeset or hand-drawn) inline or as floats;**
- **upload and submit your PDFs to Blackboard before class time on the due date.**

Always show your work and prove your constructions correct.

1. Given a string x in $\{0,1\}^*$, we define $n_0(x)$ and $n_1(x)$ to be the numbers of 0's and 1's in x respectively. Use the Pumping Lemma to prove that each of the following languages over $\{0,1\}$ is non-regular.

   (a) $\{0^n 1^m 0^{n+m} \mid m,n \geq 0\}$

   (b) $\{xy \mid |x| = |y| \text{ and } n_0(x) = n_0(y)\}$

   (c) $\{x \mid n_0(x) > 0, n_1(x) > 0, \text{ and } \mathrm{GCD}(n_0(x), n_1(x)) > 1\}$
   (*Hint*: recall that $\mathrm{GCD}(p,q) > 1$ iff for some $k > 1$ and $a, b > 0$, $p = ak$ and $q = bk$.)

2. Consider a language L_g over $\{0,1\}^*$ whose members are all strings of the form $0^n 1^{g(n)}$, where g is some integer-valued function of n. Prove that L_g is regular iff there exists an integer k such that, for all sufficiently large n, $g(n+k) = g(n)$. (*Hint*: the Pumping Lemma is useful for one direction.)

3. Let $\Sigma = \{0,1\}$, and let

$$L = \{x0a \mid x \in \Sigma^*, a \in \Sigma\}.$$

   In other words, L is the set of all bit strings whose second-to-last character is a 0.

   (a) What are the equivalence classes of I_L? Enumerate these classes, show that your enumeration is correct (complete and without duplicates), and use it to construct a a minimum-sized DFA accepting L.

   (b) What is the smallest (in terms of number of states) NFA that accepts L? Show that there are at least three *nonisomorphic* NFAs of this minimum size that all accept L.

4. For each of the following languages L, either show that I_L has infinitely many distinct equivalence classes (and hence is not regular), or show that it has finitely many distinct equivalence classes that cover all of Σ^* (and hence is regular). In the former case, you need not describe *all* equivalence classes of I_L, just show that there are infinitely many of them. In the latter case, enumerate all the classes, prove that they are all distinct, and show that they cover Σ^*.

   (a) $\{x \in \{0,1\}^* \mid x = ww\}$

   (b) $\{x \in \{0,1\}^* \mid x = wcw, |c| > 0, |w| > 0\}$

   (c) $\{x \in \{0,1\}^* \mid x = wcw^R, |c| > 0, |w| > 0\}$

5. A common extension found in practical regular expression implementations is the *backreference* operator. A (much simplified) definition of this extension is as follows. Let *ordinary* regular expressions be those we defined previously, using only concatenation, finite union, and Kleene closure. An *augmented* regular expression is either

   - an ordinary regular expression;
   - the finite union, concatenation, or Kleene closure of augmented regular expressions; or
   - $\$R\$(S)\beta$, where R is an ordinary regular expression; S is an augmented regular expression; and β is a backreference to the *delimited* expression R.

   When we match an augmented regular expression against a string, a backreference β matches only a substring identical to that which was matched by its corresponding delimited expression. For example,

$$\$(00+11)\$(0+1)^*\beta$$

   matches bit strings that begin with either 00 or 11 and end with the same two characters that they begin with. That is, 00100 is matched, but 00111 is not.

(a) Does the set of augmented regular expressions still describe the regular languages? Prove that it does or give a counterexample to show that it does not.

(b) Suppose we restrict the above definition so that delimited expressions for backreferences may not contain the Kleene closure operator $*$. Now does the set of augmented expressions still describe the regular languages? Prove that it does or give a counterexample to show that it does not.