CS581 – Theory of Computation – HW3

Tuesday, April 16, 2013
due in class Tuesday, April 23, 2013

Answer each question below. You will turn this homework in using D2L. In addition, you may also turn in a paper copy in class. In this case the TA will mark up your homework with comments and return the comments to you.

You may format your answers using some document processing software, or you may write it up with pencil and paper and scan it. In either case submit a pdf document. Be sure your submission is clearly identified as Homework 3, and contains your name and your email on the first line. The first line should look like:

CS581 HW #3 Tom Smith tsmith@pdx.edu

1. Give regular expressions for each of the languages over the alphabet $\{0, 1\}$ below.
   - $\{w | w$ begins with a 1 and ends with a 1$\}$
   - $\{w | w$ contains at least three 1's$\}$
   - $\{w | w$ has length at least three, and its third symbol is a 0$\}$
   - $\{w | w$ has a 1 in every odd position$\}$
   - $\{w | w$ contains at least two 0's and at most one 1$\}$
   - all strings except the empty string (ϵ).
   - $\{\epsilon, 0\}$
2. NFA $\rightarrow$ DFA. Sipser exercise 1.16 (page 86). Use the construction given in Theorem 1.39 to convert the following two NFA to equivalent DFA.

	1 start final	2
a	$\{1, 2\}$	$\emptyset$
b	$\{2\}$	$\{1\}$

	1	2	3
	start	final	
a	$\{3\}$	$\{1\}$	$\{2\}$
b	$\emptyset$	$\emptyset$	$\{3,2\}$
ϵ	$\{2\}$		

See the textbook for state diagrams of the two NFA. (15 points)

3. RegExp $\rightarrow$ NFA. Translate the following regular expression into an NFA. $(ac + b^*c)^*$ Show enough steps to convince me you are following the steps of Lemma 1.55 (page 67). (15 points)

4. NFA $\rightarrow$ RegExp . Translate the following NFA over the alphabet $\{a, b, c\}$ into a regular expression. Use the algorithm emobodied in Lemma 1.60 (page 69-74). (15 points)

	0	1	2
	start	final	final
a	$\{1\}$	$\emptyset$	$\emptyset$
b	$\emptyset$	$\{0,1\}$	$\emptyset$
c	$\{2\}$	$\emptyset$	$\emptyset$

Show the steps of the algorithm by drawing the original NFA, and each intermediate Generalized NFA until you have eliminated all but two states.

5. **Pumping Lemma** A pumping lemma argument that some language is not regular is always a proof by contradiction. Clearly state what you are assuming to be true. Clearly state how that leads to a contradiction. Restate what the contradiction shows to be true. Each proof should include re-stating the pumping lemma specialized to the proof you are constructing.

   Show that the language $\{xyx | x \in \{0,1\}^+, y \in \{0,1\}^*\}$ is not regular (20 points).

6. The perfect shuffle of two strings $A = x_1x_2\cdots x_n$ and $B = y_1y_2\cdots y_n$ is the string

   - $x_1y_1x_2y_2\cdots x_ny_n$ **where** n is the length of both A and B.
   - thus `(shuffle "abc" "123")` is `"a1b2c3"`

   If M and N are regular languages show that $\{\text{shuffle } x\ y | x \in M, y \in N, \|x\| = \|y\|\}$ is a regular language. (Hint: think about some sort of product construction that interleaves the steps of two different DFA). (15 points)