CS311 – Computational Structures – HW4

Tuesday, October 16, 2012
due in class Tuesday, October 23, 2012

Answer each question below. Write your answers neatly on paper. Be sure your name is on the paper, and the paper is clearly identified as Homework 4.

1. RegExp → NFA. Translate the following regular expression into an NFA. $(ac+b^*)^*$ Show enough steps to identify what algorithm you are using. (15 points)

2. NFA → RegExp . Translate the following NFA over the alphabet $\{a, b, c\}$ into a regular expression. Use the algorithm described in box 11.5 page (713). We called this Alg. 4 in the notes of October 9, 2012. (15 points)

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

3. Problem 2, Section 11.3, page 743. Lambda Closures. Build a table for each part. Clearly identify the "unions". (20 points)

4. RegExp → NFA → DFA → MinDFA. Transform the following regular expression into a minimal DFA: a^*b^*. Use the following steps:

   - RegExp → NFA. Algorithm 11.7, page 729.
   - NFA → DFA. Algorithm 11.8, pages 733-734.
   - DFA → minDFA. Algorithm 11.10, page 740, or one of the algorithms discussed in class. Show your work so we can tell what algorithm you are using.

   Show enough details to convince me you know the algorithms. (20 points.)

5. A Binary Tree (BT) is defined (inductively) as follows

   - single node, R, (with no children) is a BT
   - if x and y are BTs then (N x y) is a BT with subtrees x and y.

   Two functions on BTs are edgeCount and nodeCount which have the following properties

```
1  edgeCount(R) = 0
2  edgeCount(N t1 t2) = 2 + edgeCount(t1) + edgeCount(t2)

3  nodeCount(R) = 1
4  nodeCount(N m n) = 1 + nodeCount(m) + nodeCount(n)
```

- Draw a picture of the tree represented by (N R (N R R)). Draw another picture of a tree of your choice. Write down how it would be represented. Label both trees with their edgeCount and nodeCount. (4 points)
- Prove by structural induction that for all BT t the following property holds: *edgeCount(t) = nodeCount(t) - 1.*
  - What is the induction variable? (1 points)
  - What is the formula as a function of the induction variable. (1 points)
  - Use the defintion of BT to help you formulate the stucture of the proof (One case for each way you can make a BT). Write down the parts of the proof based on the structure of BT. (3 points) Which cases have induction hypotheses?
  - Carry out the steps of the proof, label each step with the properties you use. (3 points)
  - Write down any facts about arithmetic that you use in your proof. (3 points)

(15 points)

6. The perfect shuffle of two strings $A = x_1 x_2 \cdots x_n$ and $B = y_1 y_2 \cdots y_n$ is the string

- $x_1 y_1 x_2 y_2 \cdots x_n y_n$ **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). (15 points)