CS 311: Computational Structures **Midterm Exam** **October 30, 2012**

This exam has 5 questions. It is a closed book exam. Answer all 5 questions in the space provided.

Student Name (neatly in block letters): ______________________________

1. **Proof by induction**. (20 points) In homework this week you showed by induction, that for one representation of the natural numbers, and for one inductive definition of addition, that addition was associative, that is $i + (j + k) = (i + j) + k$. In this problem we use the same representation of the natural numbers and addition, but instead show that zero (`Z`) is a right identify for plus (`+`), that is for all x the following holds $x + Z = x$. Recall the definitions for the natural numbers and addition (infix `+`).

   - Z is a natural number (zero)
   - if n is a natural number, then $S\ n$ is a natural number (the successor function).
   - Nothing else is a natural number

   ```
   (1)  Z + n = n
   (2)  (S m) + n = S(m + n)
   ```

   Prove by structural induction that for all x the following property holds: $x + Z = x$

   (a) What is the formula as a function of the induction variable. (2 points)

   (b) Use the definition you gave in (a) above, and the definition of natural numbers (one case for each way you can make a natural number) to formulate the structure of the proof. Write this formulation down. (4 points)

   (c) Write down the parts of the proof based on the structure given in part (b) above. (2 points)

   (d) Which cases have induction hypotheses? (2 points)

   (e) Carry out the steps of the proof, label each step with the properties you use. (10 points)

2. **Short answers**. Answer each question with a sentence or two (2 points each).

   (a) What is the definition of a regular language?

   (b) List at least three ways to describe languages that describe exactly the regular languages.

   (c) Describe a technique for demonstrating that a language is not regular.

   (d) Describe an informal but concrete language that is not regular but is context free.

   (e) Give a Context free language description of the language you gave as an answer in part (d) above.

   (f) List four closure properties of the regular languages.

   (g) The "power set (set of all subsets) construction" is an important part of an algorithm for describing why two different kinds of languages are equivalent. What are those two kinds of languages?

   (h) Give the definition of an Λ-NFA as a 5-tuple. Be sure and give a label to each of the 5 parts and give a description each of the 5 parts.

(i) Give a regular expression for the language $\{w \in \{a, b, c\}^* \mid$ every a in w is immediately followed by at least one

(j) Some claim that the algorithm to convert a DFA into and NFA is trivial. Why might they say that?

3. **Binary Addition**. (20 points)

In lecture we have spent some time discussing the Mealy machine for binary addition. This machine assumes numbers are represented as a string with the lowest order bits first. The Mealy machine has two states, representing the value of the carry. The input is pairs of binary digits, one from each of the two numbers being added. We write a pair (a,b) vertically as $\begin{bmatrix} a \\ b \end{bmatrix}$

(a) Draw the Mealy machine for addition. (4 points)

(b) Give the input string for this machine representing the sum $3 + 4$ as 4 bit numbers. Give the corresponding output. (2 points)

(c) The rest of this problem constructs an NFA to recognize the language of correct binary sums. The input will be a string of triples of bits, corresponding to the input and output of the Mealy machine. So if the Mealy machine on the pairs of input $\begin{bmatrix} a_1 \dots a_n \\ b_1 \dots b_n \end{bmatrix}$ generated output $s_1 \dots s_n$, then the NFA should accept triples of input $\begin{bmatrix} a_1 \dots a_n \\ b_1 \dots b_n \\ s_1 \dots s_n \end{bmatrix}$. Note as a consequence of this specification that partial sums that generate a carry should be accepted by the machine.
All strings that do not correspond to correct sums should be rejected.

For each of the following indicate (with yes or no) if the string should be accepted or rejected: (1 point each)

i. $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ ____________

ii. $\begin{bmatrix} 10 \\ 10 \\ 01 \end{bmatrix}$ ____________

iii. $\begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix}$ ____________

iv. $\begin{bmatrix} 10 \\ 01 \\ 11 \end{bmatrix}$ ____________

(d) Define an NFA accepting this language by drawing a cartoon of the state machine. (10 points) [Hint: my solution looks very similar to the Mealy machine.]

4. **Pumping lemma** (20 points)

   (a) State the pumping lemma for regular languages. (6 points)

   (b) Describe how the pumping lemma is used to show a language is not regular. (2 points)

   (c) Use the pumping lemma to show that $\{0^i 1^j 0^k | k = i + j\}$ is not regular. (12 points)

5. **Regular expression to NFA construction.** 20 points

   Use one of the algorithms described in the text or in the class notes to construct an NFA which describes the same languages as the regular expression $c^*a(\Lambda + b)$

   (a) Use the rules of precedence to fully parenthesize the regular expression. (4 points)

   (b) Draw a graphical representaion (i.e. a tree) of your expression from (a) above. (2 points)

   (c) Show each step of the construction for the algorithm you choose. (14 points)