CSE 547T Intro. to Formal Languages and Automata	**Spring 2012**
Exam I	
Given: 21 February 2012	Due: 24 February 2012, 2 PM, Bryan 509

This exam is open-book and open-notes (your notes, not somebody else's). If you use other material in solving this exam, you must properly cite such material where used. You are not to discuss this exam with anybody except the professor.

Where possible, answer questions on the pages of the exam. Additional ruled pages are included at the end of the exam. Partial credit will be given where sufficient detail is provided.

You must sign the pledge below for your exam to count. Any cheating will cause the students involved to receive an F for this course. Other action may be taken.

Remember to state any theorem you choose to prove; for inductive proofs, be sure to provide the induction hypothesis.

To turn in the exam, you can bring it to class on Thursday, or you can bring it to Bryan 509 before 2 PM on Friday. Do not slide it under the door; hand it into the office staff when the office is open (9-5). Do not email me your exam! I need it on paper, turned in as specified above.

Name:		
Student ID:		
Problem Number	Possible Points	Received Points
1	5	
2	10	
3	25	
4	30	
5	10	
6	20	
Total	100	

Pledge: Any resources I have used to solve this exam other than my book or my notes are properly cited on the pages where I used them. On my honor, I have neither given nor received any unauthorized aid on this exam.

Signed: ______________________________

1. (5 points) Fill in the blank:

   (a) (1 points) A *language* is a set of __________.

   (b) (1 points) The smallest language is __________.

   (c) (1 points) The smallest string is __________.

   (d) (1 points) For every nonempty alphabet Σ, the set Σ^* always contains the string __________.

   (e) (1 points) For every regular set, there is a __________ that recognizes the language. Also, each regular language can be characterized by a __________ expression. Also, a specialized grammar can generate such languages, namely a __________ grammar.

2. (10 points) Describe how to solve each of the following problems. No proof is necessary, but you must provide a decision process that terminates with the right answer.

   (a) (5 points) Given two regular expressions R and S, are their corresponding languages (exactly) the same?

   (b) (5 points) Given two deterministic finite state machines M_1 and M_2, are there any strings accepted by both?

   Each problem page of this exam is followed by a ruled page for you to write your answers. Use the blank space at the bottom of this page and the ruled page. If you need more space, use the ruled pages at the end of the exam.

3. (25 points) In this problem you will investigate whether binary addition can be characterized as a regular language.

Consider the addition of two binary strings w_1 and w_2 to form w_3:

$$\begin{array}{rcl} w_1 & = & 0\ 1\ 0\ 1 \\ +\ w_2 & = & \underline{0\ 1\ 1\ 0} \\ & & \\ w_3 & = & 1\ 0\ 1\ 1 \end{array}$$

In decimal, we would write: $5 + 6 = 11$

Now suppose we vertically slice the strings w_1, w_2, and w_3 to obtain:

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

We then encode all possible slices so that

$$0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, 1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, 2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, 3 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, 4 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, 5 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, 6 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, 7 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

With this encoding, the example above is encoded (from left to right) as

1 6 3 5

but this string is processed from right to left (*i.e.*, the way we do addition). You will design an FSA that processes such strings as depicted below:

$$1\ 6\ 3\ 5 \rightarrow \boxed{\text{FSA}}$$

Your FSA must accept any string in $\{0, 1, 2, 3, 4, 5, 6, 7\}^\star$ that represents correct addition in progress. For the above string, the light is on before and after each digit that is processed. The light is initially on (no mistake has been made). The 5 is processed, and the light stays on, because $1 + 0 = 1$. The 3 is next processed, and the light stays on, because with no carry from the previous addition, $0 + 1 = 1$. After the 6, the light stays on, because $1 + 1 = 0$, with a carry, but the addition in progress is still correct. The 1 is processed, and with the carry into that computation $0 + 0 = 1$, so the light stays on.

As another example, consider $1\ 6\ 1\ 5 \rightarrow \boxed{\text{FSA}}$. The light is initially on, and stays on when the 5 is processed. But the 1 causes the light to go off and stay off, since $0 + 0 \neq 1$ (there was no carry from the first addition of $1 + 0 = 1$.

(a) (15 points) Draw a complete deterministic FSA (all transitions defined on all states) for this language. Be sure to label the start state and any accepting states appropriately.

(b) (10 points) If your FSA has n states, then develop a set of n pairwise-distinguishable strings with respect to this language. If you are unable to do this, it is likely the case that your machine is too big, so try making it smaller.

4. (30 points) While we usually think of doing addition from right to left, we next consider processing the encodings developed in Problem 3 from left to right.

   To support this, we need to build an FSA R that accepts the *reverse* of the strings accepted by your FSA in Problem 3. It will be helpful to define formally how the string w^r is constructed as the reverse of the string w:

$$\lambda^r \quad = \quad \lambda \tag{1}$$
$$w \;=\; x\;a \quad \Longrightarrow \quad w^r = a\;x^r \tag{2}$$
$$w \;=\; a\;x \quad \Longrightarrow \quad w^r = x^r\;a \tag{3}$$

   Consider a (deterministic and complete) DFA $M = (Q, \Sigma, \delta, q_0, A)$.

   (a) (10 points) Define a corresponding NFA $M^r = (Q^r, \Sigma, \delta^r, q_0^r, A^r)$ that accepts the string w^r if and only if M accepts w. Note that you may use nondeterminism and λ in this automaton construction.

   (b) (10 points) Prove that if M accepts w, then your M^r accepts w^r. Using the notation from class, this means you must prove:

$$\hat{\delta}(q_0, w) \in A \Longrightarrow \hat{\delta^r}(q_0^r, w^r) \in A^r$$

   (c) (10 points) Prove that if your M^r accepts w^r, then M accepts w:

$$\hat{\delta}(q_0, w) \in A \Longleftarrow \hat{\delta^r}(q_0^r, w^r) \in A^r$$

5. (10 points) Based on your solution to Problem 3 and your construction in Problem 4:

   (a) (5 points) Draw an NFA that accepts strings that represent correct additions in progress, processed from left to right (in *reverse* of the order symbols were processed in Problem 3).

   (b) (5 points) Using the techniques we learned in class, show a corresponding DFA for your NFA. Partial credit will be awarded only if your work is shown.

6. (20 points) Each of the languages described below is *not* regular. We learned two techniques in class for proving languages non-regular (pumping lemma and infinite set of pairwise-distinguishable strings). **To receive full credit for this problem, you must use each technique, but it is up to you to decide which technique to use on which problem.**

   You must provide the following details in your proofs:

   Pumping lemma You must begin with a string $z = uvw$ that is in the language, and the length of z must be at least n, the number of states of a hypothetical FSA. Your argument must allow that uv can occur anywhere in the first n symbols of z, and your proof must include any possible construction of v as allowed by the pumping lemma. You must show that for a particular value of i, which you disclose to us clearly, that uv^iw is not in the language.

   You must use the above notatation or we will not give you credit! This seems harsh, but we have many papers to grade, and this will simplify things for us.

   Infinite set of pairwise-distinguisable strings You must define an infinite set of strings and call it S. You must prove that for any pair of strings taken from S, calling them x and y, that there exists a z such that xz is in the language, but yz is not.

   The alphabet for each language is $\Sigma = \{ a, b \}$.

   (a) $L_a = \{ x \mid |x| = k!,\ k \geq 0 \}$. This is the set of all strings whose length is $k!$ for some k. L_a includes $\{ a, b, aababa \}$.

   (b) $L_b = \{ x \mid |x| = 2^k - 1,\ k \geq 0 \}$. This is the set of all strings whose length is one less than an integral power of 2. L_b includes $\{ b, aba, aabaaba \}$.