Midterm Exam I
CIS 341: Introduction to Logic and Automata — Fall 2003, evening
Prof. Marvin K. Nakayama

Print Name (family name first): ____

I have read and understand all of the instructions below, and I will obey the Academic Honor Code.

____

Signature and Date

- This exam has 6 pages in total, numbered 1 to 6. Make sure your exam has all the pages.
- This exam will be 1 hour and 25 minutes in length.
- This is a closed-book, closed-note exam.
- For all problems, follow these instructions:
  1. Give only your answers in the spaces provided. I will only grade what you put in the answer space, and I will take off points for any scratch work in the answer space. Use the scratch-work area to work out your answers before filling in the answer space.
  2. FA stands for finite automaton; TG stands for transition graph.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step.

Problem	1	2	3	4	Total
Points					

1. **[30 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — If S is any set of strings, then S^* is infinite.

(b) TRUE FALSE — The language Palindrome over the alphabet $\Sigma = \{a, b\}$ is closed under concatenation.

(c) TRUE FALSE — The language $L = \{a^n b^n : n \geq 0\}$ has regular expression $\mathbf{a}^*\mathbf{b}^*$.

(d) TRUE FALSE — If a language L has a regular expression, then L must be finite.

(e) TRUE FALSE — If a finite automaton has exactly one final state, then the language of the FA consists of exactly one word.

(f) TRUE FALSE — Given a transition graph T and a string w, if there is at least one way of processing w on T such that T crashes, then T rejects w.

(g) TRUE FALSE — If a language L is infinite, then it does not have a regular expression.

(h) TRUE FALSE — Every transition graph is also a finite automaton.

(i) TRUE FALSE — The regular expression $(\mathbf{a} + \mathbf{b})^*$ generates every possible string over the alphabet $\Sigma = \{a, b, c\}$.

(j) TRUE FALSE — The regular expression $(\mathbf{aa} + \mathbf{bb})^*$ generates the language over the alphabet $\Sigma = \{a, b\}$ of strings having an even number of a's and an even number of b's.

2. **[20 points]** For each of the following languages L over the alphabet $\Sigma = \{a, b\}$, give a finite automaton that accepts exactly L.

   (a) L exactly consists of all strings with an odd number of b's.

   Draw finite automaton here:

   (b) L exactly consists of all strings that do not contain the substring aab.

   Draw finite automaton here:

Scratch-work area

3. **[30 points]** Suppose we define a restricted version of the C++ programming language in which variable names must satisfy all of the following conditions:

- A variable name can only use lower-case Roman letters (i.e., a, b, ..., z) or Arabic numerals (i.e., 0, 1, 2, ..., 9). Underscore and upper-case Roman letters are not allowed.
- A variable name must start with a lower-case Roman letter.
- The length of a variable name must be no greater than 4.
- A variable name cannot be a keyword (e.g., if). The set of keywords is finite.

Let L be the set of all valid variable names in our restricted version of C++. Let L_0 be the set of strings satisfying the first 3 conditions above; i.e., we do not require the last condition.

(a) Give a regular expression for L_0.

(b) Give a formula for $|L_0|$. You do not need to give a single number.

(c) Prove that L has a regular expression, where L is the set of strings satisfying all four conditions. You do not have to give the regular expression.

Scratch-work area

4. **[20 points]** For a language L, define the transpose of L, denoted by L^t, to be the language of exactly those words in L spelled backwards; i.e., $L^t = \{\text{reverse}(w) : w \in L\}$. For example, if $L = \{a, abb, bbaab\}$, then $L^t = \{a, bba, baabb\}$. Prove that $(L_1L_2)^t = L_2^tL_1^t$ for any languages L_1 and L_2.