Midterm Exam
CS 341-451: Foundations of Computer Science II — **Fall 2014, eLearning section**
Prof. Marvin K. Nakayama

Print family (or last) name: ____________________________________

Print given (or first) name: ____________________________________

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

____

Signature and Date

- This exam has 10 pages in total, numbered 1 to 10. Make sure your exam has all the pages.
- Unless other prior arrangements have been made with the professor, the exam is to last 2.5 hours and is to be given on Saturday, October 18, 2014, 9:30am–12:00pm.
- This is a closed-book, closed-note exam. No electronic devices (e.g., calculators, cellphones) are allowed.
- 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 or the backs of the sheets to work out your answers before filling in the answer space.
  2. DFA stands for deterministic finite automaton; NFA stands for nondeterministic finite automaton; CFG stands for context-free grammar; PDA stands for pushdown automaton. TM stands for Turing machine.
  3. For any proofs, be sure to provide a step-by-step argument, with justifications for every step. If you are asked to prove a result X, in your proof of X, you may use any other result Y without proving Y. However, make it clear what the other result Y is that you are using; e.g., write something like, "By the result that $A^{**} = A^*$, we know that ...."

Problem	1	2	3	4	5	6	Total
Points							

1. **[20 points]** For each of the following, circle TRUE if the statement is correct. Otherwise, circle FALSE

(a) TRUE FALSE — A language L has a CFG if and only if L is recognized by a PDA.

(b) TRUE FALSE — If a language A is not regular, then it must be infinite.

(c) TRUE FALSE — If a language is infinite, then it must not be regular.

(d) TRUE FALSE — If A is a regular language, then A has a regular expression.

(e) TRUE FALSE — A regular expression for the language $\{\, 0^n 1^n \mid n \geq 0 \,\}$ is $0^* 1^*$.

(f) TRUE FALSE — If $A = \{01, 1\}$ and $B = \{\varepsilon\}$, then $A \times B = A \circ B$.

(g) TRUE FALSE — The class of regular languages is closed under complementation.

(h) TRUE FALSE — If A is a context-free language, then A is recognized by an NFA.

(i) TRUE FALSE — If A and B are context-free languages, then so is $A \circ B$.

(j) TRUE FALSE — If A and B are context-free languages, then so is $A \cup B$.

2. **[30 points]** Define L to be the set of strings that represent numbers in a modified version of Java. The goal in this problem is to define a regular expression and an NFA for L. To precisely define L, let $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{.\}$, where $\Sigma_1 = \{ \texttt{0}, \texttt{1}, \texttt{2}, \ldots, \texttt{9} \}$ is the set of *digits* and $\Sigma_2 = \{ \texttt{+}, \texttt{-} \}$ is the set of *signs*. Then $L = L_1 \cup L_2$, where

   - L_1 is the set of all strings that are decimal integer numbers. Specifically, L_1 consists of strings that start with an optional sign, followed by one or more digits. Examples of strings in L_1 are "`02`", "`+9`", and "`-241`".
   - L_2 is the set of all strings that are floating-point numbers that are not in exponential notation. Specifically, L_2 consists of strings that start with an optional sign, followed by zero or more digits, followed by a decimal point, and end with zero or more digits, where there must be at least one digit in the string. Examples of strings in L_2 are "`13.231`", "`-28.`" and "`.124`". All strings in L_2 have exactly one decimal point.

   Assume that there is no limit on the number of digits in a string in L. Also, we do not allow exponential notation, nor do we allow for the suffixes `L`, `l`, `F`, `f`, `D`, `d`, at the end of numbers to denote types (long integers, floats, and doubles); these symbols are not in Σ anyways.

   (a) Give a regular expression for L_1.

   (b) Give an NFA for L_1 over the alphabet Σ.

(c) Give a DFA for L_1 over the alphabet Σ. Your DFA must include all transitions.

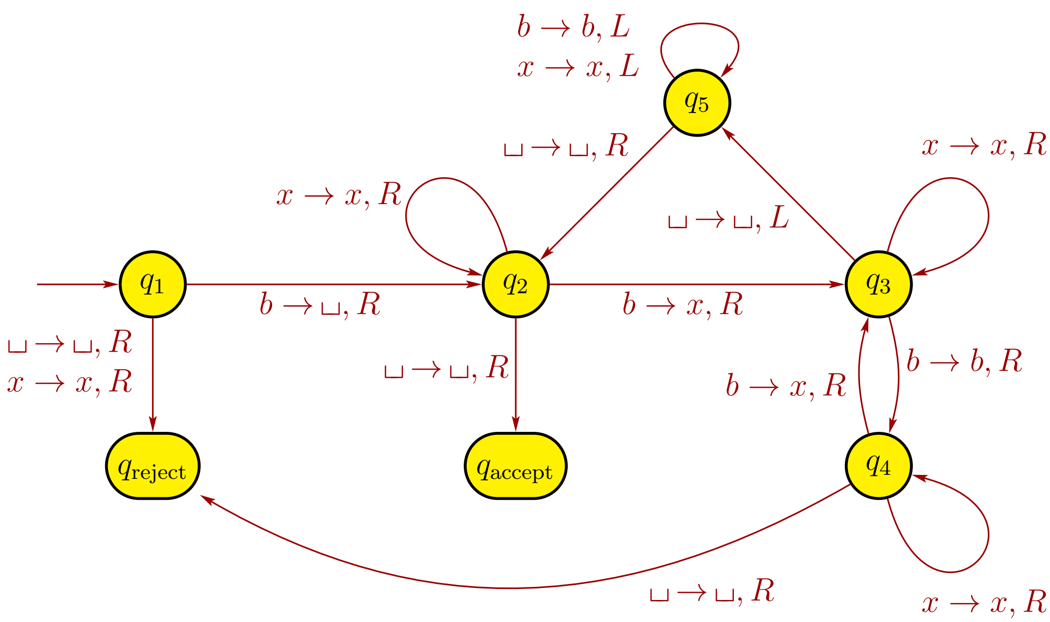
(d) Give a regular expression for L_2.

(e) Give an NFA for L_2 over the alphabet Σ.

(f) Give a regular expression for L.

(g) Give an NFA for L over the alphabet Σ.

3. **[10 points]** The Turing machine M below has input alphabet $\Sigma = \{b\}$ and tape alphabet $\Gamma = \{b, x, \sqcup\}$.



In each of the parts below, give the sequence of configurations that M enters when started on the indicated input string.

(a) bb

(b) $bbbbb$

Scratch-work area

4. **[20 points]** Let $\Sigma = \{a, b, c\}$, and consider the language $A = \{a^n b^n c^k \mid n, k \geq 0\}$.

(a) Give a CFG G for A. Be sure to specify G as a 4-tuple $G = (V, \Sigma, R, S)$.

(b) Give a PDA for A. You only need to give the drawing.

Scratch-work area

5. [**10 points**] Give an example of context-free languages A and B such that $C = A \cap B$ is not context-free. Explain your answer. Be sure to give CFGs for A and B. You do not have to prove that C is non-context-free for your example, but C must be a non-context-free language that we went over in the course.

6. **[10 points]** Recall the pumping lemma for regular languages:

Theorem: If L is a regular language, then there is a number p (pumping length) where, if $s \in L$ with $|s| \geq p$, then s can be split into 3 pieces, $s = xyz$, satisfying conditions

(i) $xy^iz \in L$ for each $i \geq 0$,

(ii) $|y| > 0$, and

(iii) $|xy| \leq p$.

Let $\Sigma = \{a, b, c\}$, and consider the language $A = \{a^n b^n c^k \mid n, k \geq 0\}$. Prove that A is not a regular language.