

CS 154 (Section 1) Formal Languages & Computability	Spring 2009 Midterm 1 (1 page(s) total)
--	--

Cell phones, pagers, etc. should all be off! No tools (calculators, computers, sliderules, screwdrivers, friends, books, or notes) should be used for the test, other than a pen/pencil, your wits and knowledge.

For this exam, you may leave up to 2 of the first 5 questions blank, and up to 2 of the last 5 (sub)questions blank. Remember how tests are scored: a blank answer is worth 30% credit. (You may also draw a large X through your scratchwork to indicate that you would prefer to take the 30%.) Spend your time working on questions which you think you can answer correctly. Good luck.

Some useful facts:

- Closure laws seen: if L_x and L_y are regular languages, then so are $L_x \cap L_y, L_x \cup L_y, L_x \cdot L_y, L_x^*, \overline{L_x}, L_x^R, L_x \ominus L_y, L_x - L_y$, and $h(L_x)$ for homomorphism $h(\)$.
 - The regular language pumping lemma: If L is a regular language, there exists some positive integer m such that any $w \in L$ with $|w| \geq m$ can be decomposed as $w = xyz$ with $|xy| \leq m$ and $|y| \geq 1$ such that $w_i = xy^iz$ is also in L for all non-negative integers i .
 - We have seen that the following languages are **not** regular: $\{a^n b^n : n \geq 0\}, \{w w^R : w \in \{a, b\}^*\}, \{w w : w \in \{a, b\}^*\}, \{a^n : n \text{ is a perfect square}\}$. You may use the fact that these languages are not regular without proof.
1. (10 points) Consider the regular expression TO BE FILLED IN. Use the rules we have seen in class to draw an NFA for it. Do not improvise, use the rules to construct the machine. I should be able to determine, from your work, what you did to get the $+$, \cdot , and $*$ operators. (An “improvised” machine is wrong, even if it happens to accept the correct language.)
 2. (10 points) For the following NFA, use the rules we have seen in class to construct an equivalent DFA. You do not need to include any states which are inaccessible by the DFA start state, but do not try to minimize the machine, or alter it, in other ways. Name states as needed to refer back to the original machine.
 3. (10 points) Consider the GTG below. Give the GTG which results after eliminating state X .
 4. (10 points) Minimize the DFA below. Name states in the minimized machine based on the states they encompass from the original machine.
 5. (10 points) Consider the following right-linear grammar, with start variable S . Construct the corresponding NFA for it, following the rules we have seen for the conversion.
 6. (10 points) Prove that $L = \text{TO BE FILLED IN}$ is not regular, using closure laws.
 7. (10 points) Prove that $L = \text{TO BE FILLED IN}$ is not regular, using the regular language pumping lemma.
 8. (10 points) Consider a language L GIVEN IN SOME WAY. Describe a procedure to accept SOME VARIATION OF L 'S LANGUAGE, TO BE FILLED IN.
 9. For each of the following languages, indicate whether or not it is regular or not. Justify your answers: for regular languages, give a regular expression, DFA, NFA, or right-linear grammar. For non-regular languages, use the pumping lemma or closure rules to prove it. (Each problem is worth 10 points, 7 of which are for the proof.)
 - (a) (10 points) LANGUAGE HERE
 - (b) (10 points) LANGUAGE HERE