

CS 154 (Section 2) Formal Languages & Computability	Spring 2009 Final Exam (3 page(s) total)
--	---

Some useful facts:

- Regular language closure laws we have 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 , and $L_x - L_y$.
- 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 .
- You may use the fact that the following languages **are not** regular, but **are** context-free, without proof:
 - $\{a^n b^n : n \geq 0\}$
 - $\{ww^R : w \in \{a, b\}^*\}$
- Context-free language closure laws we have seen: if L_x and L_y are context-free languages, then so are $L_x \cup L_y$, $L_x \cdot L_y$, L_x^* , L_x^R , and $L_x \cap L_z$ if L_z is regular.
- Context-Free Language Pumping Lemma (we have seen it, but you don't need it for this exam): If L is a context-free language, there exists some positive integer m such that any $w \in L$ with $|w| \geq m$ can be decomposed as $w = uvxyz$ with $|vxy| \leq m$ and $|vy| \geq 1$ such that $w_i = uv^i xy^i z$ is also in L for all non-negative integers i .
- You may use the fact that the following languages are **not** context-free, but **are** recursive, without proof:
 - $\{a^n b^n c^n : n \geq 0\}$
 - $\{ww : w \in \{a, b\}^*\}$
 - $\{a^n b^m a^n b^m : n \geq 0\}$
 - $\{a^n : n \text{ is a prime number}\}$
- The following problems are not recursive: The halting problem and Post's Correspondence Problem. Both are recursively enumerable.
- If a language L is R.E. and so is \overline{L} , then L is recursive.
- If language A is not recursive, but A turing reduces to B , then language B is also not recursive.
- Rice's Theorem: Consider the language $L = \{M \mid M, \text{ for Turing machine } M, \text{ accepts a language with some } \textit{non-trivial} \text{ property}\}$. L is not recursive. (Non-trivial means there exists at least one Turing machine that has a language with the property, and at least one other that has a language without the property.)

1 Regular and Context Free Languages

You may take the default 30% for *at most two of the four* questions in this section. (You may cross out a question to indicate that I should give you 30% for that one.)

1. (10 points) Give a deterministic finite automata which accepts the same language as the following non-deterministic one. You do not *need* to show steps if your final result is correct, but if you think you may have made a mistake along the way, intermediate results will help for partial credit. You may ignore the garbage state if you like. States should be named as appropriate from our algorithm.
2. (10 points) Minimize the following deterministic finite automata. You do not *need* to show steps if your final result is correct, but if you think you may have made a mistake along the way, intermediate results will help for partial credit. States should be named as appropriate from our algorithm.
3. (10 points) Give a regular expression for the following regular grammar (starting variable S):
4. (10 points) Tell how to generate the string “some string here” from the following grammar:
SOME GRAMMAR HERE
Also, describe the language.

2 Turing Machines

You may take the default 30% for *at most two of the four* questions in this section. (You may cross out a question to indicate that I should give you 30% for that one.) *Your Turing machines should be one tape machines. You may use stationary moves if you like. In general, the simpler the machine, the better (as long as it is correct).*

1. (10 points) Give an enumerator for the following language: LANGUAGE HERE
2. (10 points) Give a Turing machine which, given INPUT HERE computes FUNCTION OF INPUT HERE. Your machine should “return” its answer by entering a final state on the leftmost character of the returned string.
3. (10 points) Given the following Turing machine, give a computation (instantaneous descriptions) for what it does on input INPUT HERE.
4. (10 points) A binary encoding of a Turing machine, using our universal Turing machine protocol (from class and the text), follows. Draw the diagram for the corresponding Turing machine, using \square to denote a blank, and letters of the alphabet to denote other symbols. (As stated in class, below I assume that the string need not have a 0 “separator” at the end.)
SOME BINARY STRING HERE

3 Other

You may take the default 30% for *at most one of the two* questions in this section.

- (10 points) Consider the following instance of Post's Correspondence Problem:
 $w_1 = \text{SOME}$, $w_2 = \text{STUFF}$, $w_3 = \text{HERE}$
 $v_1 = \text{OTHER}$, $v_2 = \text{STRINGS}$, $v_3 = \text{HERE}$
Does it have a solution? (If so, give one, if not, explain why not.)
- (10 points) SOMETHING RELATED TO CLOSURE OF TURING MACHINE LANGUAGES
HERE.

4 No Choice

Answer all three questions in this section.

- (10 points) Prove that the following language is *not recursively enumerable*: DESCRIPTION HERE.
IT MIGHT BE A NOT CO-R.E. PROOF INSTEAD.
- (30 points) For each of the following six languages tell whether the language is (i) regular, (ii) context-free but not regular, (iii) recursive but not context-free, (iv) recursively enumerable but not recursive, (v) co-recursively enumerable but not recursive, or (vi) neither recursively enumerable nor co-recursively enumerable. **Proofs will be ignored, this is multiple choice from the 6 choices above.**
SIX DIFFERENT LANGUAGE DESCRIPTIONS GO HERE
- (20 points) Prove that the language $L = \text{SOME HARDER THAN USUAL LANGUAGE HERE}$ is not regular, using closure, pumping lemma, fundamentals, or any combination.