CS 273, Fall 2006
Mock/Practice Final Exam

This practice exam is based on the final from spring 2006. I've tried to modify it to reflect changes in what the two courses covered (including slightly different text/notation), but beware that I might not have fixed all divergences, especially in the first two questions. It is approximately the length you should expect for the final.

The specific choice of topics for the long problems may be different. Other especially likely choices include:

- a pumping lemma proof
- construct a PDA, Turing machine, or context-free grammar for some specified language
- given the state diagram for some machine, tell us what language it recognizes; given a set of grammar rules, tell us what language they generate
- something like problem 5, in which you describe a general procedure for modifying a class of machines, but using a PDA, NFA/DFA, or a context-free grammar

Problem 1: T/F (15 points)

Completely write out "True" if the statement is necessarily true. Otherwise, completely write "False". For example, "$x+y > x$" has answer "False" assuming that y could be 0 or negative. But "If x and y are natural numbers, then $x + y \geq x$" has answer "True". You do not need to explain or prove your answers. Your score is the number of correct answers.

1. If A is regular and B is context-free, then $A - B$ is context-free.
2. If L is not context-free and F is a finite language, then $L - F$ is not context-free.
3. Let M be a PDA whose stack has size bounded by k for some constant k. Then $L(M)$ must be regular.
4. Every regular language without epsilon has a context-free grammar in Chomsky normal form.
5. The regular languages denoted by the regular expressions $1^*(10 + 1^*)$ and $1^* + 1^*0$ are equal.
6. If L is regular, so is $\text{perm}(L) = \{w : \exists u \in L,\ w \text{ is a permutation of the characters of } u\}$.
7. If L is recursively enumerable, and $\overline{L}$ is not recursively enumerable, then L cannot be recursive.
8. If L is not recursively enumerable, then L cannot be context-free.
9. If there is an enumerator for L, then L must be recursive.
10. If there is an NFA for L, then there is also a deterministic PDA for L.
11. If L is accepted by a one-state PDA by null stack, then L is accepted by an NFA.
12. It is decidable whether or not a TM with blank input ever writes a nonblank character on its tape.
13. If L is not regular, then there is a PDA for L.
14. Both recursive and recursively enumerable languages are closed under reversal.
15. There is a universal TM M such that for all L, $L(M) = L$.

Problem 2: Classification (15 points)

In each of the following, information about a language L is given.
Circle either **R, C, REC, RE, N** depending on which of the following holds:

R: Any language satisfying the information must be regular.

C: Any language satisfying the information must be context-free, but not all languages satisfying the information are regular.

REC: Any language satisfying the information must be recursive, but not all languages satisfying the information are context-free.

RE: Any language satisfying the information must be recursively enumerable, but not all languages satifying the information are recursive.

N: Not all languages satisfying the information are recursively enumerable.

Be sure to erase completely. Ambiguously marked papers will not receive credit. Your score is the number correct.

1. **R C REC RE N** $L = \{\langle G \rangle \# w : \text{G encodes a CFG, and } w \in L(G)\}$
2. **R C REC RE N** $L = A - \overline{B}$, where A is recursive, and B is a CFL.
3. **R C REC RE N** L is accepted by a TM that always moves to the right.
4. **R C REC RE N** $L = A \cap B$, where both A and B are CFLs.
5. **R C REC RE N** $L = \{i : M_i \text{ accepts the empty string in at most 273 steps}\}$.
6. **R C REC RE N** L is the complement of a recursively enumerable language.
7. **R C REC RE N** $L = A \cap B$ where both A and B are recursively enumerable.
8. **R C REC RE N** L is a language that is accepted by both a PDA and a TM.
9. **R C REC RE N** $L = \{i : M_i \text{ accepts at most 273 strings}\}$.
10. **R C REC RE N** $L = \{i \# j : j \in L(M_i) \text{ and } i \in L(M_j)\}$.
11. **R C REC RE N** L is a subset of a regular language.
12. **R C REC RE N** $L = A \cap B$ where A is regular and B is a CFL.
13. **R C REC RE N** $L = R - S$, where R is regular and S is recursive.
14. **R C REC RE N** $L = \{n : \text{there are less than } n \text{ electrons in the universe}\}$.
15. **R C REC RE N** $L = \{0^i 1^j 0^k : i - j \neq k\}$.

Problem 3: Short answer (10 points)

(a) Suppose $010011q_31000 \vdash_M 01001q_210000$ for TM M. Then (fill in the blanks):

If δ is the transition function for M, then we can conclude that $\delta(a, b) = (c, d, e)$, where:

$a =$ ____ $b =$ ____

$c =$ ____ $d =$ ____ $e =$ ____

(b) True or False, and give a brief argument: The cardinality of the set of natural numbers exceeds the cardinality of the set of TM codes.

(c) Consider the problem of deciding whether a Turing machine M ever moves left on an input string w. Explain briefly why this problem is decidable

Problem 4: Closure (10 points)

The *quotient* of two languages A and B is defined as follows:

$A/B = \{x : \exists y \in B \text{ such that } xy \in A\}$

Intuitively, A/B can be viewed as the first parts of words in A whose last parts are in B.
For example, if $A = \{1, 001, 010\}$ and $B = \{1, 00\}$, then $A/B = \{\epsilon, 00\}$.

Prove that recursively enumerable languages are closed under quotient. That is, prove that if A and B are both recursively enumerable, then so is A/B.

Problem 5: TM variations (10 points)

A *unistate TM* is a TM with a single state q, and two tapes. A unistate TM M is said to accept a string w if when M is started with w at the left of tape 1, and with tape 2 initially blank, M eventually writes the six- character word "accept" on tape 1.

Prove that if M is any TM, then there is a unistate TM M' such that $L(M') = L(M)$. (Hint: the tape alphabet of M' will contain extra symbols.)

(a) First, give the basic idea, but giving enough details so that it is clear you understand what the key issues are.

Next, give the following details:

(b) What is the tape alphabet of M' in terms of the description of M? ________________

Let δ be the transition function of M, and δ' be the transition function of M', recalling that M' is a two-tape machine. Suppose that in M, $\delta(p, a) = (p', b, D)$ where $D \in \{L, R, S\}$ indicates that M moves left, right, or remains in one place, respectively. Then your construction of M' should somehow simulate this transition of M. Below, give values of the place-holder keywords to describe how your simulation works.

(c) For the above transition of M, we create the transition

δ'(state, tape1sym, tape2sym) = (new-state, new-tape1sym, new-tape2sym, direction1, direction2)

where (fill in the blanks)

state	= ______	tape1sym	= ______	tape2sym	= ______
new-state	= ______	new-tape1sym	= ______	new-tape2sym	= ______
		direction1	= ______	direction2	= ______

Problem 6: Parse Trees (10 points)

Let G be a grammar in Chomsky Normal Form.
Prove by induction on n, that every parse tree for a word $w \in L(G)$ of length n must have $2n - 1$ internal nodes. (Internal nodes are those that are labeled with nonterminals - hence, nodes that are not leaves of the parse tree.)

Problem 7: Regular expressions (10 points)

If S is a language, define suffixes$(S) = \{y : \exists x \in \Sigma^* \text{ such that } xy \in S\}$.

It is not hard to show that regular languages are closed under the suffix operation. That is, if S is regular, then so is suffixes(S).

Suppose r, s, r', and s' are regular expressions denoting R, S, suffixes(R), and suffixes(S), respectively.

Give regular expressions for the following languages, using only r, s, r', and s', and the standard regular expression operations (concatenation, star, and plus).

(a) suffixes$(R \cup S) =$

(b) suffixes$(RS) =$

(c) suffixes$(R^*) =$

Problem 8: Reduction (10 points)

Let $CFG_{TM} = \{< M > | M$ is a TM and $L(M)$ is a CFL$\}$.

Use a reduction to show that CFG_{TM} is undecidable. You may not use Rice's theorem.