CS3133
FINAL EXAM

Name_____

Date: October 14, 2010
All documentation permitted

1. (25 points) Give a grammar in Chomsky Normal Form to generate

$$\left\{ xyx^{rev} \mid x \in \{a,b\}^* , y \in \{c\}^* \right\} - \{\varepsilon\},$$

where x^{rev} is x written in reverse. So *abbcccbba* belongs to the language, but *acbcbca* does not.

2. (25 points) Let L be the set of all binary strings x such that every prefix of x has at least as many 0's as 1's. So $00110001 \in L$, but $001110001 \notin L$ because the prefix 00111 has more 1's than 0's. Is L a CFL? Justify your answer.

3. (25 points) The *reversal* of a language L is the language consisting of the reversal of every string in L. So the reversal of {1101, 11*abc*, ε, *helloworld*} is {1011, *cba*11, ε, *dlrowolleh*}. Are the context-free languages closed under reversal? Justify your answer.

4. (25 points) Is the following set of Turing machines recursively enumerable?

$$\left\{ M \mid \text{there is an input such that } M \text{ halts in fewer than } 453^{453} \text{ steps} \right\}$$

Justify your answer.

CS3133

Solutions to Final Exam

1. A CFG to generate $\left\{xyx^{rev} \mid x \in \{a,b\}^*, y \in \{c\}^*\right\}$ is

$$S \to aSa \mid bSb \mid A$$
$$A \to cA \mid \varepsilon$$

.

We can convert it to a CNF grammar to generate $\left\{xyx^{rev} \mid x \in \{a,b\}^*, y \in \{c\}^*\right\} - \{\varepsilon\}$:

$$S \to X_a Y_a \mid X_b Y_b \mid X_a X_a \mid X_b X_b \mid X_c A \mid c$$
$$Y_a \to SX_a$$
$$Y_b \to SX_b$$
$$A \to X_c A \mid c$$
$$X_a \to a$$
$$X_b \to b$$
$$X_c \to c$$

2. L is context-free because it is accepted by the DFA which pushes 0's onto the stack, and whenever it reads a 1 with a 0 on top of the stack it pops the 0. It is accepted by empty stack by the PDA with transition function:

$$\Delta(s,0,\perp) = \{(s,0\perp)\}$$
$$\Delta(s,0,0) = \{(s,00)\}$$
$$\Delta(s,1,0) = \{(s,\varepsilon)\}$$
$$\Delta(s,\varepsilon,\perp) = \Delta(s,\varepsilon,0) = \{(s,\varepsilon)\}$$

3. The CFLs are closed under reversal. If L is a CFL, then it is generated by a CFG G. For each production $A \to u_1 ... u_k$, $u_1,...,u_k \in N \cup \Sigma$ of G, replace it by the production $A \to u_k ... u_1$ The new grammar generates the reversal of L.

4. The language is r.e. To enumerate it, we dovetail over all the machines and inputs

	y_0	y_1	y_2	y_3	$y_4 \ldots$ (inputs)
M_0	1	2	4	7	11
M_1	3	5	8	12	
M_2	6	9	13		
M_3	10	14			

When we process the pair (M_i, y_j) we run M_i on input y_j for 453^{453} steps. If M_i halts on input y_j, we output M_i and never try it again on another input.