**Automata Qual Exam (Spring 2014)**
Answer ALL questions (Closed Book Exam)

**1.** (15 points)

We define that a language $L$ is co-Turing-recognizable if and only if the complement of $L$ is Turing-recognizable. Note that a Turing-recognizable language is also called a recursively enumerable language.

Show that the equivalence problem of context-free grammars $EQ_{\text{CFG}} = \{ <G, H> \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}$ is co-Turing-recognizable.

Hint: You can make use of the result (without proof) that the CFG membership problem $\{<G, x> \mid G \text{ is a CFG and } x \in L(G)\}$ is decidable.

Note: $<G, H>$ denotes the encoding of the grammars $G$ and $H$, and $<G, x>$ denotes the encoding of the grammar $G$ and the input $x$.

*Answer: To recognize the complement of $EQ_{\text{CFG}}$, we design a Turing machine to enumerate all strings lexicographically; and, for each string $x$ enumerated, if $x \in L(G)$ and $x \notin L(H)$, or $x \notin L(G)$ and $x \in L(H)$, then accept.*

**2.** (20 points)

Let $L$ be an infinite Turing-recognizable (recursively enumerable) language. Let $M$ be a Turing machine that accepts $L$. Explain how one can modify $M$ to return one string in $L$. Note that it does not matter which string in $L$ is returned. It is required that the Turing machine constructed must halt and return a string in $L$. That is, you cannot construct a Turing machine that may run forever without returning a string.

*Answer: Let the strings over $\Sigma^*$ be ordered lexicographically, and are named $w_1, w_2, \ldots$ respectively. We run $M$ in a time sharing way on $w_1$ for one step, on $w_2$ for one step, on $w_1$ (resume) for one step, on $w_2$ (resume) for one step, on $w_3$ for one step, on $w_1$ (resume) for one step, on $w_2$ (resume) for one step, on $w_3$ (resume) for one step, on $w_4$ for one step,... etc. When some string is accepted by $M$, then we return that string.*

**3.**

Let $L = \{baba^2ba^3 \ldots ba^{n-1}ba^nb \mid n \geq 1\} \subseteq \Sigma^*$ where $\Sigma = \{a, b\}$.

(a) (15 points) Show that $L$ is not context-free using the pumping method.

*Answer (Sketch): Let $p$ be the pumping constant. Then consider $baba^2ba^3 \dots ba^{p-1}ba^pb$. For all different ways of breaking the string into $u, v, w, x, y$, we pump the string down which results in a string not in $L$.*

(b) (20 points) Show that $\Sigma^* - L$ (i.e., the complement of $L$) is context-free.

*Answer: We design a nondeterministic PDA to accept strings in $\Sigma^* - L$. The PDA checks, using the finite state control, that the string begins with bab and ends with b. If not, PDA accepts. Also, the PDA nondeterministically guesses a substring $ba^iba^jb$ to check if $j = i + 1$ by first pushing $i$ copies of $a$ into the stack, which later are matched against $j$ copies of $a$ that are read next. If it is found that $j \neq i + 1$, then PDA also accepts.*

**4.**

Given a string $w$, we define its reversal $w^R$ inductively as follows: $\epsilon^R = \epsilon$ and $(xa)^R = a(x^R)$, where $a \in \Sigma$ and $x \in \Sigma^*$.

For a language $L$, we write $L^R = \{w^R \mid w \in L\}$.

(a) (10 points) Show how to define *formally*, for each NFA $M = (Q, \Sigma, \delta, q_0, F)$, an NFA $M'$ such that $L(M') = L(M)^R$. Note that we do not allow an NFA to have $\epsilon$ transitions. But an NFA is allowed to have multiple starting states.

*Answer: $M' = (Q, \Sigma, \delta', F, \{q_0\})$ such that $(p, a, q) \in \delta'$ iff $(q, a, p) \in \delta$.*

(b) (20 points) Explain how to prove *formally* that your construction of $M'$ is correct. Note: you do not have to provide a detailed formal proof; but a careful explanation of how the formal proof is organized, and on what principles that proof is based is expected.

*Answer: Recall that $\delta \subseteq Q \times \Sigma \times Q$. We define $\delta^* \subseteq Q \times \Sigma^* \times Q$ inductively such that $(p, a, q) \in \delta^*$ if $(p, a, q) \in \delta$, and $(p, xy, q) \in \delta^*$ if $(p, x, q') \in \delta^*$ and $(q', y, q) \in \delta^*$. Similarly, we define $\delta'^*$ given $\delta'$. Using induction on the the structures of $\delta^*$ and $\delta'^*$, we show that for every string $x$, $(p, w, q) \in \delta^*$ iff $(q, w^R, p) \in \delta'^*$. To show that $L(M') = L(M)^R$, we verify that for all $w$, $w \in L(M')$ iff $\exists q \in F$, $(q, w, q_0) \in \delta'^*$ iff $\exists q \in F$, $(q_0, w^R, q) \in \delta^*$ iff $w^R \in L(M)$ iff $w \in L(M)^R$.*