Automata Qual Exam (Spring 2012)
Answer ALL questions (Closed Book Exam)

Question 1 (15 points)

(a) If $L_1 \bigcup L_2$ is regular, then L_1 is regular.
Answer: No. Let $L_1 = \{a^n b^n \mid n \geq 0\}$ and $L_2 = \{a, b\}^*$.
(b) If $L_1 \cdot L_2$ is regular, then L_1 is regular.
Answer: No. Let $L_1 = \{a^n b^n \mid n \geq 0\}$ and $L_2 = \{a, b\}^*$.
(c) If L^* is regular, then L is regular.
Answer: No. Let $L = \{a^n b^n \mid n \geq 0\} \cup \{a, b\}$.

Question 2

Consider the following context-free grammar G:

$$S \longrightarrow aaSb \mid aSbb \mid \epsilon$$

Note: $L(G) \subseteq a^* b^*$. Below are the possible i and j such that $a^i b^j \in L(G)$:

i	j
0	0
1	2
2	1, 4
3	3, 6
4	2, 5, 8
5	4, 7, 10
6	3, 6, 9, 12
7	5, 8, 11, 14
8	4, 7, 10, 13, 16
9	6, 9, 12, 15, 18
10	5, 8, 11, 14, 17, 20
11	7, 10, 13, 16, 19, 22
12	...
13	...
...	...

(a) (15 points)

It is given that $L(G) = \{a^{2n} b^{f(n,k)} \mid 0 \leq k \leq n\} \cup \{a^{2n+1} b^{g(n,k)} \mid 0 \leq k \leq n\}$.
What are $f(n, k)$ and $g(n, k)$?

Answer: $f(n,k) = n + 3k$ and $g(n,k) = n + 3k + 2$.

(b) (15 points) Prove that the characterization for $L(G)$ given in part (a) is correct using mathematical induction. Note: you can assume *without proof* that $L(G) \subseteq a^*b^*$.

Answer:
Let $L_n = \{a^{2n}b^{n+3k}, a^{2n+1}b^{n+3k+2} \mid 0 \leq k \leq n\}$. Claim: $L(G) = \cup_{n\geq 0} L_n$.

As it is assumed that $L(G) \subseteq a^*b^*$, and L_n's differ in the number of a's, it suffices to prove by induction on n that $L(G) \cap (a^{2n}b^* \cup a^{2n+1}b^*) = L_n$.

Base case ($n = 0$)
From the grammar rules, it is clear that $L(G) \cap (a^0b^* \cup a^1b^*) = \{a^0b^0, a^1b^2\}$.
On the other hand, $L_0 = \{a^0b^{3k}, a^1b^{3k+2} \mid 0 \leq k \leq 0\} = \{a^0b^0, a^1b^2\}$.

Induction hypothesis: ($n = p$)
It is assumed that $L(G) \cap (a^{2p}b^* \cup a^{2p+1}b^*) = L_p$.

Induction step: ($n = p + 1$)

(1) To show $L(G) \cap a^{2(p+1)}b^* = \{a^{2(p+1)}b^{(p+1)+3k} \mid 0 \leq k \leq (p+1)\}$.

There are two cases in the derivation sequence for a string with $2(p+1)$ a's.

Case (i) $S \to aaSb \overset{*}{\Rightarrow} aawb = \alpha$ where $w \in a^{2p}b^j$.
Since $S \overset{*}{\Rightarrow} w = a^{2p}b^j$ and by the induction hypothesis, $j = p + 3k$ for $0 \leq k \leq p$. Therefore, $\alpha = a^{2p+2}b^{j+1} = a^{2(p+1)}b^{(p+1)+3k}$ for $0 \leq k \leq p$.

Case (ii) $S \to aSbb \overset{*}{\Rightarrow} awbb = \beta$ where $w \in a^{2p+1}b^j$.
Since $S \overset{*}{\Rightarrow} w = a^{2p+1}b^j$ and by the induction hypothesis, $j = p + 3k + 2$ for $0 \leq k \leq p$. Therefore, $\beta = a^{2p+2}b^{j+2} = a^{2(p+1)}b^{p+3k+2+2} = a^{2(p+1)}b^{(p+1)+3k+3} = a^{2(p+1)}b^{(p+1)+3(k+1)}$ for $0 \leq k \leq p$. Equivalently, $\beta = a^{2(p+1)}b^{(p+1)+3k}$ for $1 \leq k \leq p + 1$.

The two cases give rise to strings $a^{2(p+1)}b^{(p+1)+3k}$ for $0 \leq k \leq p + 1$.

(2) To show $L(G) \cap a^{2(p+1)+1}b^* = \{a^{2(p+1)+1}b^{(p+1)+3k+2} \mid 0 \leq k \leq (p+1)\}$.

There are two cases for deriving a string with $2(p + 1) + 1$ a's.

Case (i) $S \to aaSb \overset{*}{\Rightarrow} aawb = \alpha$ where $w \in a^{2p+1}b^j$.
Since $S \overset{*}{\Rightarrow} w = a^{2p+1}b^j$ and by the induction hypothesis, $j = p+3k+2$ for $0 \leq k \leq p$. Therefore, $\alpha = a^{2p+3}b^{j+1} = a^{2(p+1)+1}b^{(p+1)+3k+2}$ for $0 \leq k \leq p$.

Case (ii) $S \to aSbb \stackrel{*}{\Rightarrow} awbb = \beta$ where $w \in a^{2(p+1)}b^j$.
Since $S \stackrel{*}{\Rightarrow} w = a^{2(p+1)}b^j$ and by the result of (1), $j = (p+1) + 3k$ for $0 \leq k \leq (p+1)$. Therefore, $\beta = a^{2(p+1)+1}b^{j+2} = a^{2(p+1)+1}b^{(p+1)+3k+2}$ for $0 \leq k \leq p+1$.

The two cases give rise to strings $a^{2(p+1)+1}b^{(p+1)+3k+2}$ for $0 \leq k \leq p+1$.

(c) (10 points) Give a context-free grammar G' such that $L(G') = \{w \mid w \in L(G),\ |w| \text{ is even } \}$.

Answer: $S_e \to aaS_eb \mid aaS_ebbbb \mid \epsilon$

(d) (10 points) Give a context-free grammar G'' such that $L(G'') = \{w \mid w \in L(G),\ |w| \text{ is odd } \}$.

Answer: $S_o \to aS_ebb$

Question 3

(a) (20 points) Explain how a deterministic Turing machine can simulate a nondeterministic Turing machine for recognizing the same language.

Answer: (see the textbook)

(b) (15 points) Suppose we modify the definition of nondeterministic Turing machine so that a string is accepted if the string is accepted by every possible computation path. (In contrast, a normal nondeterministic Turing machine accepts a string w if there exists one accepting path that accepts w.) Explain how a deterministic Turing machine can simulate a nondeterministic Turing machine according to the modified definition.

Answer: Let w be the input to a nondeterministic Turing machine M. If M accepts w, M will accept w in all possible computation paths. That is, there is a time t such that all computation paths are completed successfully within time t. We want to simulate M by a deterministic machine M'. M' will try out each possible t in ascending order. For a specific t, M' will enumerate all possible computation paths of lengths within t in a lexicographical way according to the sequence of 'choices' made during the nondeterministic computation. For a specific t, there are finitely many enumeration of paths of lengths at most t. Therefore, M' can try out all the computation paths in a finite amount of time. If all computation paths are successfully completed within t steps, then M' accepts. If there are some computation paths that fail within t steps, then M' rejects. Otherwise, every computation path may either succeed or may attempt to continue for more than t steps. In the last case, M' will start another round of simulation for computation paths of lengths up to $t+1$ steps.