Solution sketch 3 - Computational Models - Spring 2013

1. (a) False. $L_1 = \{0^p | p \in \text{Primes}\}$ $\quad L_2 = 0^*$ $\quad x = 00000000$ $\quad y = 00$
   (b) False. $x \in L_1L_2/L_2 \iff \exists y \in L_2 \;\; xy \in L_1L_2$ but by (a) this does not necesarily implies that $x \in L1$.
   (c) 0^*11^*
   (d) ϕ (a word in $(01)^*$ can't end with 0)
   (e) $\{0^n1^m | m \leq n\}$
   (f) $L_1 = \{a^ib^jc^k\#c^kb^ja^i | i, j, k \in N\}, \quad L_2 = \{\#c^nb^na^n | n \in N\}$

2. (a) CFG. $S \rightarrow aaSbbb|\epsilon$
   (b) Not CFG. Pump as follows: Given n, choose $w = 0^n1^{n^2} \in L$. Now, for a partition $w = uvxyz$ such that $|uvx| \leq n$ and $n \geq k = |vy| \geq 1$ and vy contains only 0 letters we choose $i = 2$ and $uv^2xy^2z = 0^{n+k}1^{n^2} \notin L$ since $n + k < (n + 1)^2$. for a partition $w = uvxyz$ such that $|uvx| \leq n$ and $n \geq k = |vy| \geq 1$ and the number of 1 letters in vy is t we choose again $i = 2$ and uv^2xy^2z has $n+t$ letters 1 which again cant be asquare number of ones.
   (c) L_3 is not context free. Proof by the pumping lemma. Let $l > 0$ be the critical length. Choose $s = 0^l1^l\#0^l1^l$. $|s| \geq l$ and $s \in L_3$. Assume $s = uvxyz$, $|vy| > 0$, $|vxy| \leq l$:
      - If vy contains # then for every $i \neq 1$, $uv^ixy^iz \notin L_3$.
      - If vy is in the first part (before the #) then for every $i > 1$, $uv^ixy^iz \notin L_3$.
      - If vy is in the second part (after the #) then for $i = 0$, $uv^ixy^iz \notin L_3$.
      - Otherwise, vy contains 1's from the first part and 0's from the second part (why?). Then for $i = 2$, $uv^ixy^iz \notin L_3$ (why?).
   (d) CFG. $S \rightarrow 0S0|1S1|0T1|1T0 \;\; T \rightarrow 0T|1T|\epsilon$

3. (a) If $\exists w.|w| > n$ then as we learned in the pumping lemma, this word can be pumped infinitely and therefore $L(A)$ is infinite. if $L(A)$ is infinite, then there is a word $w \in L(A)$ such that $|w| > n$. The run of this word in A contains a cycle. We remove all cycles from the run and remember one simple cycle c, $|c| \leq n$. The run without the cycles give a word $w' \in L(A), |w'| < n$. We start pumping w' with the cycle c and we will eventually get a word in $L(A)$ in the proper length.

   (b) By (a) above, given a DFA A, we can run in A all the words w, such that $n < |w| \leq 2n$. If one of the words is accepted, then $L(A)$ is infinite, otherwise - finite.

   (c) For a grammar in CNF, a derivation of a word is actually a binary tree with binary internal nodes (representing rules of the form $A \to BC$) and unary leaves (the letters, representing rules of the form $A \to a$). Therefore, the tree for a derivation of a word of length n is at least $\log n$ deep (NOT INCLUDING THE LEAVES, since the last derivation, and only the last, in every path is unary). We conclude that in the derivation tree of a word of length more than 2^n (n being the number of variables in the grammar) there is a path with more than n variables, therefore a variable repeats and we may pump to get infinite many words in the language. Now, If $L(G)$ is infinite then there is a word $w \in L(G)$ such that $|w| > 2^n$. The minimum size parse tree for this word contains a variable A that appears twice on some path from the root to some leaf. We will remove all duplicate variables on the same path by shrinking as we learned in the pumping lemma, and get a word w', $|w'| \leq 2^n$. Then we start pumping A. Assuming we chose A to be the lowest variable that appear twice on some path, it is guaranteed that $|vy| \leq 2^n$ and we will eventually get a word in $L(G)$ of the proper length.

   (d) By (c) above, given a CFG G we can check if G generates any word w, such that $2^n < |w| \leq 2^{n+1}$. If one of the words is generated by G, then $L(G)$ is inifinite, otherwise - finite.