Gavin Grob
CS 510 Automata Theory
Homework 6

1. Use induction to prove that if G is a CFG in Chomsky normal form then any string w $\in L(G)$ of length $n \leq 1$ exactly $2n-1$ steps are required for derivation.

We know that in Chomsky normal form the derivation tree is a depth of n, and has at most 2n leave nodes.

Base Case:
Lets take a look at the most basic string w where length $n = 1$
$S \rightarrow a$, that took 1 step, and 2(1) - 1 = 1.

Inductive Step:
Lets assume that for any string of length $w \geq 2$. w can be split into substrings y and z, where $y \geq 1$ and $z \geq 1$ and $w = y + z$. Since $w \geq 2$, then $S \rightarrow YZ$ where Y and Z can go to either Terminals or NonTerminals, but Y leads to y, and Z leads to z. So the steps of the strings y and z are equal to $2n-1$ where n is the length of y and z accordingly. So the total number of steps to produce w is the starting step plus the steps to produce y and z. $1 + (2(L(y)) - 1) + (2(L(z)) - 1) = 2(L(x+y)) - 1 = 2(L(w)) - 1$. So (2(L(y)) - 1) and (2(L(z)) - 1) can either recursively apply this y z split for nonTerminals or it can apply the base case for Terminal states. Showing that a string of n length has $2n-1$ steps that are required for derivation.

2. Consider the following language
$L = \{a^h b^i c^j d^k | h, i, j, k \leq 0 \text{ and if } h = 1 \text{ then } i = j = k\}$
This language is not Context Free.

(a) First show that the CFL pumping lemma can not be used to prove this language is not connect free. To be perfectly clear, you have to show that any string in L can be broken up according to the rules of the pumping lemma and be pumped.

We must show that any string will not work in L, so the possible variations of strings in L are:

1. $a^1 b^z c^z d^z$
Here we want to make $i \neq j$, $i \neq k$, or $j \neq k$, any will work. The problem is that the adversary, can simple slide the window over to the section of $abbb...$ select $v = a$ and $x = \epsilon$ and if either pumped up or down, the string will stay in the language.

$2.a^h b^i c^j d^k \;\; h \neq 1$
Here we want unpump to make $a = 1$. The problem is that the adversary, can simple make vwx equal to all b's, c'c or d's and if either pumped up or down, the string will stay in the language.

$3.a^h$
This is technically case 2, but it hard to see. Here we have no chance, because $i = j = k$, so even if "somehow" we unpumped making $h = 1$ the string is in the language.

(b) Show that this language is not context free using closure properties.

So lets assume L is Context free, with this true the closure properties are Union, Concatenation, Star, String reversal, Homomorphism, and Intersection with a regular language. Lets consider the regular language $G = \{ab^p c^q d^r | p, q, r \geq 0\}$, so under the closure properties of context free languages, L $\cap$ G = H is context free. H = $\{ab^z c^z d^z | z \geq 0\}$ Because this is context free it should hold for the CFL pumping lemma.

We have a number n where vwx $\leq$ n, we then pick a string where n $\geq$ z. The available windows for the adversary to choose are:
ab..bb
bb..bb
bb..cc
cc..cc
cc..dd
dd..dd

If the adversary chooses *ab..bb* then we can simply pump/unpump what ever v and x he chooses. It will either put the b's out of sync, or he can choose the single a which can be pumped/unpumped to make more/less than a single a.

If the adversary chooses *bb..bb*, *cc..cc*, or *dd..dd* then we can simply pump/unpump what ever v and x he chooses, making $b^z c^z d^z$ not hold true. Because one of the b's, c's or d's will be out of sync.

If the adversary chooses *bb..cc* or *cc..dd* then we can simply pump/unpump what ever v and x he chooses, making $b^z c^z d^z$ not hold true. Because one or two of the b's, c's or d's will be out of sync.

Thus showing us the pumping lemma does not hold for H, so H is not Context Free. Then our assumption that L is Context free must be wrong, making L not Context Free.