(1) With $\Sigma = \{a, b, c\}$, design a context-free grammar $G$ such that $L(G)$ equals the complement of $\{a^n b^n c^n : n \geq 0\}$. (27 pts. total, 9 for the regular grammar for $\sim a^* b^* c^*$ and 18 for the rest of $G$, including the observation that the complement of $L(G)$ is not a CFL.)

Answer: First design the DFA $M$ with states $s, p, q, d$ and arcs:

$$
\begin{array}{ccc}
(s, a, s) & (s, b, p) & (s, c, q) \\
(p, a, d) & (p, b, p) & (p, c, q) \\
(q, a, d) & (q, b, d) & (q, c, q) \\
(d, a, d) & (d, b, d) & (d, c, d)
\end{array}
$$

In the complemented machine $M'$, $d$ is the only accepting state, and $L(M') = \sim a^* b^* c^*$. A regular grammar $G'$ such that $L(G') = L(M')$ is then given by:

$$
S' \rightarrow aS | bP | cQ \\
P \rightarrow aD | bP | cQ | a \\
Q \rightarrow aD | bD | cQ | a | b \\
D \rightarrow aD | bD | cD | a | b | c
$$

(Writing $A_p, A_q, A_d$ for the variables was fine too, of course. So was replacing the five terminal rules by the single rule $D \rightarrow \epsilon$. The original definition of regular grammar disallowed $\epsilon$-rules, but a rule $S \rightarrow \epsilon$ is needed as a “patch” on the definition when $\epsilon$ is in the language anyway. Note how the issues explored on problem 2 in assignment 2 resurface for grammars.)

Now let’s introduce variables $A, B, C$ that generate lists of 0-or-more $a$’s, $b$’s, and $c$’s, respectively:

$$
A \rightarrow aA | \epsilon, \quad B \rightarrow bB | \epsilon, \quad C \rightarrow cC | \epsilon
$$

and variables $T, U$ that generate $\{a^n b^n\}$ and $\{b^n c^n\}$:

$$
T \rightarrow aTb | \epsilon, \quad U \rightarrow bUc | \epsilon.
$$

Then the language $\{a^m b^n : m < n\}$ is generated by $TbB$; the terminal $b$ here ensures there is at least 1 more $b$ than $a$. Finally we put it all together by observing that in strings of the form $x = a^i b^j c^k$, $i, j, k$ are not all equal if and only if $i > j$, $i < j$, $j > k$, or $j < k$:

$$
S \rightarrow S' | aATC | TbBC | AbBU | AUCc.
$$

(Here the order of $aA, bB, \text{and/or} Cc$ can be switched around, but $T$ and $U$ must go where shown.) The grammar $G$ with start symbol $S$ then generates $L = \sim \{a^n b^n c^n : n \geq 0\}$. So $L$ is a CFL, but since its complement is $\{a^n b^n c^n : n \geq 0\}$ back again, which is not a CFL, the class of context-free languages is not closed under complementation.
Let $T$ be the language of all strings that do not have the substring $bb$. Let $G$ be the following context-free grammar:

$$
S \rightarrow \epsilon \mid b \mid AS \mid SC \\
A \rightarrow a \mid bCaA \\
C \rightarrow aS \mid ACC
$$

(a) Is $L(G) \subseteq T$? OK, it’s not. One of the rules is buggy. Fix it by deleting or changing one occurrence of one variable and call the revised grammar $G’$.

(b) Then prove that your revised grammar $G’$ is sound, i.e., $L(G) \subseteq T$. (9 + 15 = 24 pts.)

**Answer:** (a) The fault is shown by the derivation $S \Rightarrow SC \Rightarrow bC \Rightarrow bACC \Rightarrow bbCaACC$ and we’re toast. We can “blame” $S \rightarrow b$ or $A \rightarrow bCaA$ but the fixes would involve changing $b$ and the question said to change a variable occurrence. If we change $S \rightarrow SC$ then we’ll get in trouble from $S \rightarrow AS \rightarrow bCaAS \rightarrow bACCaAS \rightarrow bbCaACCaAS$ anyway. So for a single-rule change it comes down to $C \rightarrow ACC$. We can either change $A$ to $C$ or just delete $A$. The reasoning is similar in either case so let’s delete $A$ to yield the revised grammar $G’$ as:

$$
S \rightarrow \epsilon \mid b \mid AS \mid SC \\
A \rightarrow a \mid bCaA \\
C \rightarrow aS \mid CC
$$

(b) Now to prove that this is sound, first note that the variables $A$ and $C$ preserve additional properties similar to that on problem 2 of assignment 6. Namely, every string derived by $A$ must end in $a$, and every string derived by $C$ must begin with $a$. These facts help us verify that every rule prevents a substring $bb$ from occurring as follows. Since no rule has a $bb$ substring already, the only way it can occur is between two variables or $b$ and a variable on the right-hand side of a rule. Let’s check all the rules for this:

- $S \rightarrow \epsilon$, $S \rightarrow b$, $A \rightarrow a$: nothing to do.
- $S \rightarrow AS$: The substring $u$ derived by $A$ must end in $a$. Therefore the juncture between $A$ and $S$ can cause no issue.
- $S \rightarrow SC$: Now $C$ must derive a substring $v$ that begins with $a$, so again a $bb$ at the juncture is avoided.
- $A \rightarrow bCaA$: The $bC$ is no problem since eventually it will become $bv$ where $v$ begins with $a$. And the $Ca$ and $aA$ parts are innocuous because they involve an $a$.
- $C \rightarrow aS$: Again immediately innocuous.
- $C \rightarrow CC$: The first $C$ could derive a string $u$ ending in $b$ such as by $CC \rightarrow aSC \rightarrow abC$ so there is “danger,” but the second $C$ wards off the danger since the string it derives must begin with $a$. 


Therefore $L(G) \subseteq T$. (In fact, the grammar is comprehensive because $T = (a \cup ba)^*(b \cup \epsilon)$ and $G$ has the sub-derivations $S \Rightarrow AS \Rightarrow aS$ and $S \Rightarrow SC \Rightarrow bC \Rightarrow baS$. These carry out the $((a \cup ba)^*)$ part while leaving the final $(b \cup \epsilon)$ to the rules $S \rightarrow \epsilon \mid b$. FYI, this also means that the rules $A \rightarrow bCaA$ and $C \rightarrow CC$ (or $C \rightarrow CCC$ if you did that instead) are redundant and can be removed without changing the language—this is true even though those rules cannot be directly simulated by 2 or more derivation steps.)

(3) With $\Sigma = \{a, b, c\}$, define $A = \{wc^m w^R : w \in \{a, b\}^*, \; m = |w|\}$. Prove using the CFL Pumping Lemma that $A$ is not a CFL. (This is an easier version of the text problem 2.45 on page 158 (page 132 in older editions) with the string “$t$” fixed to be all-c’s. (18 pts., for 69 total on the set)

Answer: Given the (choice by the “adversary” of a) “pumping length” $N > 0$, take $x = a^Nc^N a^N$. Then $x \in A$. Let any breakdown $x = yuvwz$ such that $|uvw| \leq N$ and $uw \neq \epsilon$ be given. Then by $|uvw| \leq N$, the “compass” $uvw$ cannot touch both of the $a^N$ intervals. And by $uw \neq \epsilon$, it must touch a nonempty part of one of those intervals and/or the $c^N$ part. Hence when we “pump down” to the string $x^{(0)} = yvz$, we get $a^i c^j a^k$ where at least one of $i, k$ remains equal to $N$. If both remain equal to $N$, we must have $j < N$, which violates the “$m = |w|$” clause with $m = j$ and $w = a^N$. Else, we violate the clause that the $a^k$ part must equal the $a^i$ part reversed. So $x^{(0)} \notin A$. Since $N$ and the breakdown are general given the restrictions, this shows $A$ is not a CFL by the CFL Pumping Lemma. (“Pumping up” to $x^{(2)}$ works fine too, but you might feel you have to “pay lip service” to cases that violate the “$wc^m w^R$” format in other ways.)