1. Show that the following languages are context-free by writing down a CFG. Prove that your construction is correct.
   (a) $L = \{w \in \{0,1\}^* \mid \text{ number of } 0s \text{ and } 1s \text{ in } w \text{ are equal }\}$.
   (b) $L = \{w\#x \mid w \neq x, \text{ where } w, x \in \{0,1\}^*\}$.

Solution: Let's think about how the strings in L will look like - we can write $w = u\sigma v$ and $x = y\sigma' z$ such that one of the following holds, where $\sigma, \sigma' \in \{0,1\}$.

1. $|u| = |y|$ and $\sigma \neq \sigma'$ or
2. $|u| = |y|$ and exactly one of v or z is ε

We will cover both cases separately. The first case is as follows.

$$\begin{aligned}
S &\longrightarrow P_0 0T \mid P_1 1T \\
T &\longrightarrow 0T \mid 1T \mid \varepsilon \\
P_0 &\longrightarrow 0P_0 0 \mid 1P_0 1 \mid 0P_0 1 \mid 1P_0 0 \mid 1R \\
P_1 &\longrightarrow 0P_1 0 \mid 1P_1 1 \mid 0P_1 1 \mid 1P_1 0 \mid 0R \\
R &\longrightarrow 0R \mid 1R \mid \#
\end{aligned}$$

The non-terminal P_0 generates all strings of the form $u1v\#y$, where $|u| = |y|$, and P_1 generates all strings of the form $u0v\#y$ where $|u| = |y|$. Now, from $S \longrightarrow P_0 0T$ generates strings of the form $u1v\#y0z$ where, $|u| = |y|$, and $S \longrightarrow P_1 1T$ generates strings of the form $u0v\#y1z$ where $|u| = |y|$. Thus for every $w\#x$ such that $S \xrightarrow{\star} w\#x$, we have $w \neq x$ and $w_i \neq x_i$ for some position i.
Similarly, suppose that $w \neq x$ are such that $w_i \neq x_i$ for some i. Wlog assume that $w_i = 1$ and $x_i = 0$. Then, set $u = w_1 w_2 \ldots w_{i-1}$, $\sigma = 1$, and the remainder of w as v. Similarly, set $y = y_1 y_2 \ldots, y_{i-1}$, $\sigma' = 0$, and remainder of x as z. We can derive $w\#x$ as follows:

$$S \longrightarrow P_0 0T \xrightarrow{\star} u1v\#y0T \xrightarrow{\star} u1v\#y0z = w.$$

The second case is as follows.

$$\begin{aligned}
S &\longrightarrow V \mid TZ \\
V &\longrightarrow 0V0 \mid 0V1 \mid 1V0 \mid 1V1 \mid 0R \mid 1R \\
R &\longrightarrow 0R \mid 1R \mid \# \\
T &\longrightarrow 0T0 \mid 0T1 \mid 1T0 \mid 1T1 \mid \# \\
Z &\longrightarrow 0Z \mid 1Z \mid \varepsilon
\end{aligned}$$

Notice that V generates all strings of the form $uv\#y$, where $|u| = |y|$ and $|v| > 0$. The non-terminal T generates all strings of the form $u\#y$ where $|u| = |y|$, and Z generates all strings in $(0+1)^*$ of non-zero length. Thus $S \longrightarrow TZ$ generates all strings of the form $u\#yz$ where $|u| = |y|$ and $|z| > 0$.

(c) $L = \{0,1\}^* - \{0^n 1^n \mid n \geq 0\}$.

Solution: Every string $w \in L$ can be one of two types.

1. $w \notin L(0^*1^*)$ - since $L(0^*1^*)$ is regular, we can easily give a CFG that generates these types of strings (Do this as an exercise).

2. $w \in L = \{0^i 1^j \mid i \neq j\}$

   We can split L into two sets L_1 and L_2, where $L_1 = \{0^i 1^j \mid i > j\}$ and $L_2 = \{0^i 1^j \mid i < j\}$. We will write the grammar for L_1, and L_2 will be similar (and left as an exercise).

$$S \longrightarrow 0S1 \mid 0R$$
$$R \longrightarrow 0R \mid \varepsilon$$

Alternate concise solution: Consider the grammar

$$S \longrightarrow 0S1 \mid 1C \mid C0$$
$$C \longrightarrow 0C \mid 1C \mid \varepsilon$$

Notice that C generates $\{0,1\}^*$. All strings that start with 1 or end with 0 are clearly L, and are generated starting from the production rules $S \longrightarrow 1C$ and $S \longrightarrow C0$. Every string starting with 0 and ending with 1, will be generated starting with the production $S \longrightarrow 0S1$. But then, to generate a string in $\{0,1\}^*$ one of the productions $S \longrightarrow 1C$ or $S \longrightarrow C0$ must be used and this will generate a string that has $0^i 1^j$ where $i \neq j$ or is not of the form 0^*1^*.

(d) $L = \{a^i b^j c^k \mid i \neq j \text{ or } j \neq k\}$.

Solution: Similar to Part (c) - do it yourself.

2. Let $L \subseteq \Sigma^*$ be a context-free language with a CFG G. Write down the CFG for the language suffix(L) defined as follows: suffix$(L) = \{v \mid uv \in L \text{ for some } u \in \Sigma^*\}$.

Hint: It might help to assume that G is in Chomsky Normal Form.

Solution: Suppose that G is in Chomsky Normal Form, and every production is of the form $A \longrightarrow BC$ or $A \longrightarrow \varepsilon$. For each non-terminal A in G, we add a new non-terminal A_{suff} which will have the following property:

$$A_{\text{suff}} \xrightarrow{\star} w \Leftrightarrow \exists u \text{ s.t } A \xrightarrow{\star} uw.$$

If all productions are of the form $A \longrightarrow BC$, then recursively the suffixes of strings generated by A would be given by the rules $A_{\text{suff}} \longrightarrow B_{\text{suff}}C$ or $A_{\text{suff}} \longrightarrow C_{\text{suff}}$. Also, for $A \longrightarrow \sigma$, we need to add $A_{\text{suff}} \longrightarrow \sigma$ and $A_{\text{suff}} \longrightarrow \varepsilon$. Finally, the new start symbol will be S_{suff} (corresponding to S in G). Use induction to prove that the new grammar generates suffix(L).

We will do induction on the length of the derivation to show that $A_{\mathsf{suff}} \xrightarrow{\star} w$ iff $\exists u$ such that $A \xrightarrow{\star} uw$. For $n = 1$, $A_{\mathsf{suff}} \longrightarrow \sigma$ iff $A \longrightarrow \sigma$, and $A_{\mathsf{suff}} \longrightarrow \varepsilon$ iff $\exists \sigma$ such that $A \longrightarrow \sigma$. This proves the base case of the induction.

For the induction step, suppose that $A_{\mathsf{suff}} \xrightarrow{n} w$. There are two cases to consider depending on the first production rule that is applies on A_{suff}.

1. $A_{\mathsf{suff}} \longrightarrow C_{\mathsf{suff}} \xrightarrow{n-1} w$ for some C_{suff}: Thus, there must exists a production of the form $A \longrightarrow BC$ in G and by the induction hypothesis, there exists v such that $C \xrightarrow{n-1} vw$. Let $B \xrightarrow{\star} u$ for some u. Then $A \xrightarrow{\star} uvw$.
2. $A_{\mathsf{suff}} \longrightarrow B_{\mathsf{suff}}C \xrightarrow{n-1} w$: Suppose that we are doing a leftmost derivation, and we have $A_{\mathsf{suff}} \longrightarrow B_{\mathsf{suff}}C \xrightarrow{\star} uC \xrightarrow{\star} uv = w$. Thus $B_{\mathsf{suff}} \xrightarrow{\star} u$ and hence $\exists t$ such that $B \xrightarrow{\star} tu$, and $C \xrightarrow{\star} v$. Hence we have $A \longrightarrow BC \xrightarrow{\star} tuv = tw$.

The other direction can be proved similarly. We have assumed that for every non-terminal A, $\exists w \in \Sigma^*$ such that $A \xrightarrow{\star} w$.

3. Let $L \subseteq \Sigma^*$ be a CFL generated by the CFG G. Prove that $L^R = \{w \in \Sigma^* \mid w^R \in L\}$ is context-free by writing down the CFG generating L^R.

Solution: For every production $A \longrightarrow \alpha$ in G, add the production $A' \longrightarrow \alpha^R$ in G^R. Use induction to show that $L(G^R) = L(G)^R$.

When doing induction, take the smallest(in length) leftmost derivation and look at the last step of the derivation. You will need to use the stronger induction hypothesis about sentential forms to prove the final statement - i.e. show that $A \xrightarrow{\star} \alpha$ iff $A' \xrightarrow{\star} \alpha^R$ in G^R where α is a sentential form.

4. A *right-linear grammar* $G = (N, \Sigma, P, S)$ is one where each production in P is of the form $A \longrightarrow wB$ or $A \longrightarrow w$ where $A, B \in N$ and $w \in \Sigma^*$. In this problem, you will prove that the language generated by a right-linear grammar is regular.

(a) Show that if $G = (N, \Sigma, P, S)$ is a right-linear grammar, then there is another right-linear grammar $G' = (N', \Sigma, P', S')$ where each production in P' is of the form $A \longrightarrow \sigma B$ or $B \longrightarrow \sigma$ where $A, B \in N$ and $\sigma \in \Sigma$ such that $L(G) = L(G')$.

Solution: Add new non-terminals and split the original production, like the way we saw in class for Chomsky Normal Form.

(b) Convert G' to an NFA $N = (Q, \Sigma, \delta, q_0, F)$ such that $L(N) = L(G')$.

Solution: The states Q correspond to the non-terminals of the grammar. If $A \longrightarrow \sigma B$ is a production, then $\delta(A, \sigma) = B$. Use induction to prove the correctness of the construction.

5. A *left-linear grammar* $G = (N, \Sigma, P, S)$ is one where each production in P is of the form $A \to Bw$ or $A \to w$ where $A, B \in N$ and $w \in \Sigma^*$. Show that for every left-linear grammar G, there is a right-linear grammar G' such that $L(G) = L(G')$.

Hint: Use Problems 3 and 4 and properties of regular languages that you learnt earlier.

Solution: First, use Problem 3 to obtain the right-linear grammar for L^R. Then, use Problem 4 to obtain an NFA for L^R. Now, use the closure properties of regular languages to obtain an NFA for $(L^R)^R = L$. Now, use the construction from class to obtain the right-linear grammar for L.

6. Consider the following CFG G.

$$\begin{aligned} S &\to aAB \mid aBA \mid bAA \mid \epsilon \\ A &\to aS \mid bAAA \\ B &\to aABB \mid aBAB \mid aBBA \mid bS. \end{aligned}$$

Let $\#_a(w)$ and $\#_b(w)$ be the number of as and bs in $w \in \{a, b\}^*$. Prove using induction the following statements.

1. $S \xrightarrow{\star} w \Leftrightarrow \#_a(w) = 2\#_b(w)$,
2. $A \xrightarrow{\star} w \Leftrightarrow \#_a(w) = 2\#_b(w) + 1$, and
3. $B \xrightarrow{\star} w \Leftrightarrow \#_a(w) = 2\#_b(w) - 2$.

Use the parts above to conclude that $L(G)$ is precisely the set of strings $w \in \{a, b\}^*$ such that $\#_a(w) = 2\#_b(w)$.

Solution: For a sentential form α, let $\#_a(\alpha)$ be the number of as and As in the sententiatl form α. Similarly, define $\#_b(\alpha)$ be the number of bs and Bs in the sentential form α. Use induction suitably.