1. Give context-free grammars for the following languages:
   (a) \( \{0^n1^m0^2|n, m \geq 0\} \):
   
   \[
   S \rightarrow 0S0|T \\
   T \rightarrow 1T|\varepsilon
   \]

   (b) \( \{0^n10^m|n \leq m\} \):
   
   \[
   S \rightarrow 0S0|T \\
   T \rightarrow T0|1
   \]

   (c) \( \{c^5a^n b^m|n > m\} \):
   
   \[
   S \rightarrow cccccT \\
   T \rightarrow aTb|U \\
   U \rightarrow aU|a
   \]

   (d) \( \{a^n \#b^m \#c^m|n, m \geq 0\} \):
   
   \[
   S \rightarrow A\#T \\
   A \rightarrow aA|\varepsilon \\
   T \rightarrow bTc|\#
   \]

2. In class, we gave the following two grammars for the language
   \( L = \{w|w\text{ contains the same number of } a\text{'s and } b\text{'s}\} \).
   
   \[
   S \rightarrow \varepsilon|aB|bA \\
   A \rightarrow aS|bAA \\
   B \rightarrow bS|aBB
   \]

   and
   
   \[
   S \rightarrow aSb|bSa|SS|\varepsilon
   \]

   For each of these grammars, give both a leftmost derivation and a parse tree for the string \(aababba\).

   **Solution:**

   A leftmost derivation in the first grammar is:

   \[
   S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow aabB \Rightarrow aabS \Rightarrow aabab \\
   \Rightarrow aababS \Rightarrow aababba \Rightarrow aababbaS \Rightarrow aababba
   \]
A leftmost derivation in the second grammar is:

\[ S \Rightarrow SS \Rightarrow aSbS \Rightarrow aaSbS \Rightarrow aabbS \Rightarrow aabbSS \Rightarrow aabbaSbS \Rightarrow aabbaSbS \Rightarrow aabba \]

Parse trees corresponding to these derivations are:
3. (a) Give a context-free grammar that generates the language

\[ A = \{ w \in \{a, b\}^* | w \text{ has exactly two more } a\text{'s than } b\text{'s} \} \]

[Hint: In class, we gave two grammars that generate the set of strings with the same number of \(a\)'s as \(b\)'s. You can use either one of these grammars as part of the grammar you give.]

**Solution:**

\[
S \rightarrow TaTaT \\
T \rightarrow aTb|bTa|TT|\epsilon
\]

(b) Give a justification that your grammar is correct. Use the same type of argument as we gave in class that the grammars that generate strings with the same number of \(a\)'s and \(b\)'s are correct. You do not need to reprove the correctness of these grammars.

**Solution:** We know from class that the variable \(T\) derives exactly the strings of terminals with the same number of \(a\)'s as \(b\)'s.

Suppose now that a string \(w\) of terminals has exactly two more \(a\)'s than \(b\)'s. We need to analyze the string and see that it matches the rule with \(S\) on the left side.

We consider a counter that counts each \(a\) as \(+1\) and each \(b\) as \(-1\). At the end of the string \(w\), the count is exactly \(2\), so there must be a first place in the string where the count is \(2\). Since the count changes by only \(1\) or \(-1\) with each symbol read, there must be a place where the count reaches \(1\) and the first such place is before the first place where the count reaches \(2\).
This means we can write \( w = w_1cw_2dw_3 \), where the count first reaches 1 after reading \( c \) and the count first reaches 2 after reading \( d \). If \( c = b \), then the count is 2 at the end of \( w_1 \), contradicting that the count first reaches 2 at \( d \), so \( c = a \) and the count is 0 at the end of \( w_1 \). If \( d = b \), then the count is 3 at the end of \( w_2 \), so would have to be equal to 2 someplace before reading \( d \), which contradicts that \( d \) is the first place the count reaches 2, so \( d = a \) and the count is 1 at the end of \( w_2 \). Thus, \( w = w_1aw_2aw_3 \). Since \( w_1 \) brings the count from 0 to 0, \( w_1 \) has the same number of \( a \)'s as \( b \)'s. Since \( w_2 \) brings the count from 1 to 1, \( w_2 \) has the same number of \( a \)'s as \( b \)'s. Since \( w_3 \) brings the count from 2 to 2, \( w_3 \) has the same number of \( a \)'s as \( b \)'s. Thus, \( w_1, w_2 \) and \( w_3 \) can all be derived from \( T \) starting with the rule \( S \rightarrow TaTaT \).

(A full proof would also require showing by induction on the length of the derivation that if a string of terminals \( w \) is derivable from \( S \), then \( w \) has exactly two more \( a \)'s than \( b \)'s, but I was not asking for this.)


(This is a difficult problem, but give it a try. As a hint, you can use three variables other than the start symbol. One variable generates all strings with the same number of \( a \)'s as \( b \)'s, one variable generates strings with the same number of \( a \)'s as \( b \)'s that have the additional property that every prefix has at least as many \( a \)'s as \( b \)'s, and the third variable generates all strings with the same number of \( a \)'s as \( b \)'s that have the additional property that every prefix has at least as many \( b \)'s as \( a \)'s.)

**Solution:**

\[
S \rightarrow V aV aT \\
T \rightarrow \varepsilon | aWbT | bVaT \\
W \rightarrow \varepsilon | aWbW \\
V \rightarrow \varepsilon | bVaV
\]

5. Give right regular grammars for the following languages:

(a) \( \{w \in \{0,1\}^* \mid \text{the first and last symbols of } w \text{ are the same} \} \).

**Solution:**

\[
S \rightarrow 0|1|0T|1U \\
T \rightarrow 0T|1T|0 \\
U \rightarrow 0U|1U|1
\]

(b) \( \{w \in \{0,1\}^* \mid w \text{ contains exactly two } 1\text{'s} \} \).

**Solution:**

\[
S \rightarrow 0S|1T \\
T \rightarrow 0T|1U \\
U \rightarrow 0U|\varepsilon
\]
(c) \{w \in \{0,1\}^* | |w| \geq 3 \text{ and the third symbol from the right in } w \text{ is a } 0\}.

Solution:

\[
\begin{align*}
S & \rightarrow 0S|1S|0T \\
T & \rightarrow 0U|1U \\
U & \rightarrow 0|1
\end{align*}
\]

6. Using the method from class, convert the DFA given in the solutions to Problem 2 on Homework 1 into a right linear grammar.

Solution:

\[
\begin{align*}
A_q & \rightarrow 0A_r|1A_q|\varepsilon \\
A_r & \rightarrow 0A_r|1A_s|\varepsilon \\
A_s & \rightarrow 0A_r|1A_t|\varepsilon \\
A_t & \rightarrow 0A_t|1A_t
\end{align*}
\]

7. Using the method from class, transform the following right linear grammar into an NFA

\[
\begin{align*}
S & \rightarrow 0S|1T|\varepsilon \\
T & \rightarrow 0U|1T \\
U & \rightarrow 0U|0
\end{align*}
\]

Solution: We first replace the rule \( U \rightarrow 0 \) to obtain:

\[
\begin{align*}
S & \rightarrow 0S|1T|\varepsilon \\
T & \rightarrow 0U|1T \\
U & \rightarrow 0U|0Z \\
Z & \rightarrow \varepsilon
\end{align*}
\]

We then obtain the following NFA: