5.1 Write a Context Free Grammar for the language $L$, where
$L = \{ a^i b^j c^k | k = (i - j), \text{if } i \geq j, \text{else } k = 0 \}$. Hint: Splitting into two cases makes your job easier.

\[
S \rightarrow A | C \\
A \rightarrow aA \cup B \\
B \rightarrow aBB \cup \lambda \\
C \rightarrow aCb | Cbb \\
\]

3. Let $L_1, L_2$ be Non-Regular CFLs; $R_1, R_2$ be Regular; Answer is about $S$ and there should be just one cell per row that has an $X$.

<table>
<thead>
<tr>
<th>Definition of $S$ / Characterization of $S$</th>
<th>Always Regular</th>
<th>At worst CFL</th>
<th>Might not be CFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S = L_1 \cdot L_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$S = R_1 - L_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$S = R_1 - R_2$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$S \supseteq R_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$S \subseteq R_1$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$S = L_1 \cap R_1$</td>
<td></td>
<td>$\times$</td>
<td></td>
</tr>
</tbody>
</table>

3. Which of the following are correct definitions of an ambiguous grammar? In each case, $w$ is a terminal string. Write T(true) or F(false) in the underlined area following each statement

- There are two distinct parse trees for some string $w$ derived by the grammar $T$
- There are two distinct derivations of some string $w$ derived by the grammar $F$
- There are two distinct rightmost derivations of some string $w$ derived by the grammar $T$
- There are two distinct leftmost derivations of some string $w$ derived by the grammar $T$

4. Show that Context-Free Languages are closed under Non-Empty-Left-Right Quotient with Regular Languages. Non-Empty-Left-Right Quotient of a CFL $L$ and a Regular Language $R$, both of which are over the alphabet $\Sigma$, is denoted NELRQ($L$, $R$), and defined as

\[ \text{NELRQ}(L, R) = \{ y \mid xyz \in L; x, z \in R; \text{and } y \neq \lambda \}. \]

That is, we select a non-empty substring $y$ of $xyz$ in $L$, provided $x$ and $z$ are both in the Regular Language $R$.

You may assume substitution $f(a) = \{ a, a' \}$, and homomorphisms $g(a) = a'$ and $h(a) = a$, $h(a') = \lambda$. Here $a \in \Sigma$ and $a'$ is a distinct new character associated with each $a \in \Sigma$. No justification is required.

\[ \text{NELRQ}(L, R) = \bigcup (f(L) \cap g(R)) \Sigma^+ g(R) \]
10 5. Consider some language \( L \). For each of (a) and (b), and for each of the three possible complexities of \( L \), indicate whether this is possible (Y or N) and present evidence. Recall that

\[
\text{max}(A) = \{ w \mid w \in A \text{ and for no } x \neq \lambda \text{ does } wx \in A \}
\]

If you answer Y, you must provide an example language \( A \) and the resulting \( L \). In the case of part (b) you must also present a homomorphism \( \sigma \). If you answer N, state some known closure property that reflects a bound on the complexity of \( L \). Note: I did the first of each of the three parts for you.

a.) \( L = \text{max}(A) \) where \( A \) is context-free, not regular.

Can \( L \) be Regular? Circle Y or N.

If yes, show \( A \) and argue \( \text{max}(A) \) is Regular; if no, why not?

YES. Let \( A = \{ a^i b^i \mid i, j > 0 \text{ and } j > i \} \)

\( L = \text{max}(A) = \emptyset \), a regular set, as every string in \( A \) can be extended with more b’s.

Can \( L \) be a non-regular CFL? Circle Y or N.

If yes, show \( A \) and argue \( \text{max}(A) \) is a CFL; if no, why not?

\[
A = \sum a^n b^n \mid n > 0
\]

\( L = \text{max}(A) = A = \sum a^n b^n \mid n > 0 \)

Can \( L \) be more complex that a CFL? Circle Y or N.

If yes, show \( A \) and argue \( \text{max}(A) \) is not a CFL; if no, why not?

\[
A = \sum a^k b^c \mid k \leq i \text{ or } k \leq j
\]

\( L = \text{max}(A) = \sum a^k b^c \mid b = \text{max}(c, i) \)

\( L \) is known (proven) to be a CSL, non-CFL language.

b.) Let \( \sigma \) be a homomorphism from \( \Sigma \) into regular languages, such that, for each \( a \in \Sigma \), \( \sigma(a) = w_a \), for some string \( w_a \). Let \( A \) be a context free, non-regular language and let \( L = \sigma(A) \).

Can \( L \) be Regular? Circle Y or N.

If yes, show \( A \) and \( \sigma \), and argue \( \sigma(A) \) is Regular; if no, why not?

YES. Let \( A = \{ a^n b^n \mid n > 0 \} \)

Define \( \sigma(a) = \lambda \) and \( \sigma(b) = \lambda \)

\( L = \sigma(A) = \{ \lambda \} \), a regular set.

Can \( L \) be a non-regular CFL? Circle Y or N.

If yes, show \( A \) and \( \sigma \), and argue \( \sigma(A) \) is a CFL; if no, why not?

\[
L = \{ a^n b^n \mid n > 0 \}
\]

Define \( \sigma(a) = a \) and \( \sigma(b) = b \)

\( L = \sigma(A) = \{ a^n b^n \mid n > 0 \} \) which is a CFL.

Can \( L \) be more complex that a CFL? Circle Y or N.

If yes, show \( A \) and \( \sigma \), and argue \( \sigma(A) \) is not a CFL; if no, why not?

CFLs are known to be closed under substitution and homomorphism.
i. Use the Pumping Lemma for CFLs to show that the following language \( L \) is not Context Free.
\[
L = \{ a^n b^{\text{sum}(1..n)} \mid n > 0 \}
\]
Here \( \text{sum}(1..n) = \sum_1^n i \).
Be explicit as to why each case you analyze fails to be an instance of \( L \) and, of course, make sure your cases cover all possible circumstances. I have done the first two steps for you.

**Assume \( L \) is Context Free**

*Provides a whole number \( N > 0 \) that is the value associated with \( L \) based on the Pumping Lemma*

Choose \( z = a^N b^{(N+1)N/2} \in L \)

\[
\therefore z = \alpha \eta \theta \\
\text{with } |\alpha \theta| \leq N, (nx) > 0 \text{ AND } \\
|\eta| \geq 0 \text{ and } \alpha \eta \theta \in L
\]

\[
\exists! \quad \text{CASE 1: } \alpha \eta \theta \text{ contains only } b's.
\]

Set \( l = 2 \) then PL says

\[
a^N b^{(N+1)N/2 + (nx)} \in L
\]

but \( |nx| > 0 \) and so we have too many \( b's \), thus \( \alpha \eta \theta \not\in L \).

*Note*: Can also use \( l = 0 \).

**Case 2**: \( nx \) contains at least one \( a \)

Then \( \alpha \eta \theta \) contains at most \( n-1 \) \( b's \). Set \( l = 6 \)

In best case, we have \( n+1 \) \( a's \) and \( (n+1)N/2 + (n-1) b's \)

But \( n+1 \) \( a's \) requires \( (n+1)N/2 + (n+1) b's \)

And \( (n+1)N/2 + (n-1) < (n+1)N/2 + (n+1) \)

so \( \alpha \eta \theta \not\in L \).

As cases 1 and 2 cover all possibilities we have a contraction showing \( \alpha \eta \theta \not\in L \)

In all cases, hence \( L \) is not a CFL.
10 7. Present the CKY recognition matrix for the string aababb assuming the Chomsky Normal Form grammar, \( G = ( \{ S, T, U, V, W, A, B \}, \{ a, b \}, R, S ) \), specified by the rules \( R \):

\[
\begin{align*}
S & \rightarrow A \ T \ | \ B \ U \\
T & \rightarrow b \ | \ B \ S \ | \ A \ V \\
U & \rightarrow a \ | \ A \ S \ | \ B \ W \\
V & \rightarrow T \ T \\
W & \rightarrow U \ U \\
A & \rightarrow a \\
B & \rightarrow b
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>a</th>
<th>b</th>
<th>a</th>
<th>b</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UA</td>
<td>-UA</td>
<td>TB</td>
<td>UA</td>
<td>TB</td>
<td>TB</td>
</tr>
<tr>
<td>2</td>
<td>W</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>U</td>
<td>U</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>W</td>
<td>U</td>
<td>T</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>U</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A little help

<table>
<thead>
<tr>
<th>Non-Terminal</th>
<th>First Symbol in Rules</th>
<th>Second Symbol in Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>None</td>
<td>( T \rightarrow B \ S ; U \rightarrow A \ S )</td>
</tr>
<tr>
<td>T</td>
<td>( V \rightarrow T \ T )</td>
<td>( S \rightarrow A \ T ; V \rightarrow T \ T )</td>
</tr>
<tr>
<td>U</td>
<td>( W \rightarrow U \ U )</td>
<td>( S \rightarrow B \ U ; W \rightarrow U \ U )</td>
</tr>
<tr>
<td>V</td>
<td>None</td>
<td>( T \rightarrow A \ V )</td>
</tr>
<tr>
<td>W</td>
<td>None</td>
<td>( U \rightarrow B \ W )</td>
</tr>
<tr>
<td>A</td>
<td>( S \rightarrow A \ T ; T \rightarrow A \ V ; U \rightarrow A \ S )</td>
<td>None</td>
</tr>
<tr>
<td>B</td>
<td>( S \rightarrow B \ U ; T \rightarrow B \ S ; U \rightarrow B \ W )</td>
<td>None</td>
</tr>
</tbody>
</table>

Is aababb in \( L(G) \)? \( \checkmark \)

What is the order of execution of this approach to determine if some \( w, |w| = N \), is in \( L \)? \( N^3 \)

What is the algorithmic strategy, e.g., greedy, divide and conquer, dynamic programming, backtracking or randomized, associated with this CKY algorithm called? Dynamic Programming
Consider the CFG $G = (\{S, A, B\}, \{a, b\}, R, S)$ where $R$ is:

$S \rightarrow SAb | AbBa$

$A \rightarrow aS | a$

$B \rightarrow bS | b$

In the PDAs below, you may (and are encouraged) to use a transition diagram where transitions have arcs with labels of form $a, \alpha \rightarrow \beta$ where $a \in \Sigma \cup \{\lambda\}$, $\alpha, \beta \in \Gamma^*$. Note: This just means that you can use extended stack operations that push or pop arbitrary length strings.

a.) Present a pushdown automaton that parses the language $L(G)$ using a top down strategy.

INITIAL CONTENTS OF STACK = $S$

b.) Now, using the notation of IDS (Instantaneous Descriptions, $[q, x, z]$), describe how your PDA in (a) accepts strings generated by $G$.

$$[q_0, w, S] \xrightarrow{\lambda} [q_0, \lambda, \lambda]$$

c.) Present a pushdown automaton that parses the language $L(G)$ using a bottom up strategy. Note: I am fine with your showing strings that are on top of the stack in either reversed or non-reversed form.

INITIAL CONTENTS OF STACK = $S$

$$\lambda, bA \rightarrow S$$

$$\lambda, AB \rightarrow S$$

$$\lambda, A \rightarrow a$$

$$\lambda, a \rightarrow a$$

$$\lambda, S \rightarrow S$$

$$\lambda, \lambda \rightarrow \lambda$$

$$\lambda, b \rightarrow b$$

$$\lambda, \# \rightarrow \#$$

$$[q_0, w_j, \#] \xrightarrow{\lambda} [\lambda, \lambda, \lambda]$$

d.) Now, using the notation of IDS (Instantaneous Descriptions, $[q, x, z]$), describe how your PDA in (c) accepts strings generated by $G$. 

$$\sum_q, w_j, \# \xrightarrow{\lambda} \sum, \lambda, \lambda$$
3 a.) Consider the context-free grammar $G_1 = (\{S, A, B\}, \{0, 1\}, R_1, S)$, where $R_1$ is:

$S \rightarrow AB$
$A \rightarrow 0A0 | \lambda$
$B \rightarrow 1B1 | \lambda$

Remove all $\lambda$-rules, except possibly for a start symbol, creating an equivalent grammar $G_1'$. Show all rules.

$ Nullable = \{S, A, B\}$

$S' \rightarrow \lambda | A | B | AB$
$S \rightarrow A | B | AB$
$A \rightarrow 0A0 | 100$
$B \rightarrow 1B1 | 11$

3 b.) Consider the context-free grammar $G_2 = (\{S, A, B\}, \{0, 1\}, R_2, S)$, where $R_2$ is:

$S \rightarrow AB | B$
$A \rightarrow 1A0 | 10$
$B \rightarrow A | AA$

Remove all unit rules, creating an equivalent grammar $G_2'$. Show all rules.

$Unit(S) = \{S, B, A\}$; $Unit(A) = \{A\}$; $Unit(B) = \{B, A\}$

$S \rightarrow AB | 1A0 | 10 | AA$
$A \rightarrow 1A0 | 10$
$B \rightarrow AA | 1A0 | 10$
9. f. Consider the context-free grammar G3 = ( { S, A, B }, { 0, 1 }, R3, S ), where R3 is
   S → AB | BB
   A → 1A0
   B → 0B1 | 01

   Remove all non-productive non-terminals, creating an equivalent grammar G3’. Show all rules.
   Productive = { S, B } ; Unproductive = { A }}

   \[
   S \rightarrow BB \\
   B \rightarrow 0B1 | 01
   \]

---

9. g. Consider the reduced context-free grammar G4 = ( { S, A, B }, { 0, 1 }, R4, S ), where R4 is
   S → AABB
   A → 1B0 | 10
   B → 0A1 | 01

   Convert an equivalent grammar G4’. Show all rules.
   \[
   S \rightarrow <AAB>B \\
   A \rightarrow <1B><0> | <1><0> \\
   B \rightarrow <0A><1> | <0><1> \\
   <AAB> \rightarrow <AA>B \\
   <AA> \rightarrow AA \\
   <1B> \rightarrow <1B> \\
   <0A> \rightarrow <0>A \\
   <o> \rightarrow 0 \\
   <1> \rightarrow 1
   \]

   OR \[
   S \rightarrow <AA><BB> \\
   <AA> \rightarrow AA \\
   <BB> \rightarrow BB
   \]