1. (12 pts) Consider the following languages. Write "REG" if it is regular, "CFL" if it is a CFL and not regular, and write "NOT" if it is not a CFL.

(a) \( L = \{ a^n b^n a^n \mid n > 0 \}, \Sigma = \{a, b\} \). \( \text{NOT} \)

(b) \( L = \{ a^{2n} b^n c^n d^n \mid n > 0, p > 0 \}, \Sigma = \{a, b, c, d\} \). \( \text{NOT} \)

(c) \( L = \{ w \in \Sigma^* \mid n_a(w) < n_b(w), n_c(w) \mod 5 = 2 \}, \Sigma = \{a, b, c\} \). \( \text{CFL} \)

(d) \( L = \{ a^m b^n c^n d^n \mid m + p > n + q, m > 0, n > 0, p > 0, q > 0 \} \). \( \text{CFL} \)

(e) \( L = \{ a^m b^n c^n \mid p > m > n > 0, n < 100 \}, \Sigma = \{a, b, c\} \). \( \text{CFL} \)

(f) \( L = \{ a^m b^n c^n \mid p \mod 2 = 0, n \mod 3 = 0, n > 0, m > 0, p > 0 \} \). \( \text{REG} \)

2. (14 pts) Answer TRUE or FALSE to each of the statements below.

(a) Suppose \( R \) is a regular expression with more than 20 symbols in its alphabet. Then there exists a TM \( M \) such that \( L(M) = L(R) \) and \( M \) halts on all inputs. \( \text{TRUE} \) or \( \text{FALSE} \)?

(b) Suppose \( S \) is the start variable in a CFG and there is only one \( S \) production that is \( S \to X_1 X_2 X_3 X_4 X_5 X_6 \) where each \( X_i \in V \). Then \( \lambda \) cannot be in the FIRST(\( S \)). \( \text{TRUE} \) or \( \text{FALSE} \)?

(c) In the CFG to NPDA conversion that models LL parsing, an NPDA with three states is built. There is only one type of transition on the middle state, a transition that pops the left hand side of a production rule off the stack and pushes the right hand side of the production rule onto the stack. \( \text{TRUE} \) or \( \text{FALSE} \)?

(d) A Turing machine has an infinite tape. The tape alphabet \( \Sigma \) for a Turing machine must be finite. \( \text{TRUE} \) or \( \text{FALSE} \)?

(e) In LR parsing if the two marked rules \( B \to .a \) and \( C \to b .a \) are in the same state, then a shift/shift conflict will result, meaning the grammar is not an LR(1) grammar. \( \text{TRUE} \) or \( \text{FALSE} \)?

(f) For an L-system, the rewriting rules \( h \) are defined as \( h : \Sigma^* \to \Sigma^* \). \( \text{TRUE} \) or \( \text{FALSE} \)?
(g) The following grammar is in Greibach Normal Form. (TRUE or FALSE?)

\[
S \rightarrow aS \\
S \rightarrow bBB \\
B \rightarrow bB \mid c
\]

3. (4 pts) List the four types of actions in an LR parser.

*Shift, reduce, accept, error*

4. (6 pts) Consider the following CFG.

\[
S \rightarrow AB \\
A \rightarrow aa \\
B \rightarrow bB \mid c
\]

(a) Give the derivation of \textit{aabc} if parsed with a top-down parser.

\[
S \Rightarrow AB \Rightarrow aaB \Rightarrow aabB \Rightarrow aabc
\]

(b) Give the derivation of \textit{aabc} if parsed with a bottom-up parser.

\[
S \Rightarrow AB \Rightarrow AabB \Rightarrow Abc \Rightarrow aabc
\]

5. (4 pts) The following grammar is LL(k) for what value of \( k \)? Give the value of \( k \) and an example of two strings that need that value of \( k \) to distinguish which rule to apply.

\[
S \rightarrow ABCA \mid BaCd \\
A \rightarrow abA \mid \lambda \\
B \rightarrow b \mid ba \\
C \rightarrow c \mid ca
\]

\[
\text{start with } S \Rightarrow BaCd \\
\text{start with } S \Rightarrow ABCA
\]

\[k = 5\]
6. (10 pts) Consider the following grammar (DO NOT change the grammar):

\[
\begin{align*}
S & \rightarrow cAB \\
A & \rightarrow aA \mid \lambda \\
B & \rightarrow Bc \mid b
\end{align*}
\]

For this problem you will construct the LL(1) parse table.

(a) Calculate FIRST and FOLLOW for the variables in the grammar.

<table>
<thead>
<tr>
<th></th>
<th>FIRST</th>
<th>FOLLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>c</td>
<td>$</td>
</tr>
<tr>
<td>A</td>
<td>a, \lambda</td>
<td>b</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>c, $</td>
</tr>
</tbody>
</table>

(b) Calculate all entries in the LL(1) Parse Table. If there are multiple rules for an entry, write in all the rules.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td>cAB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>aA</td>
<td>\lambda</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>Bc</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(c) Is this grammar an LL(1) grammar? Explain. **No, needs more than 1 lookahead to decide between B \rightarrow b and B \rightarrow Bc when the lookahead is a, b.**
7. (16 pts) Construct the LR parsing table for the following grammar (DO NOT change the grammar.) A new start symbol S' and production have already been added to the grammar.

0) S' → S  
1) S → BAB  
2) A → Aa  
3) A → λ  
4) B → b  
5) B → c

(a) Calculate the FIRST and FOLLOW sets of variables.

<table>
<thead>
<tr>
<th></th>
<th>FIRST</th>
<th>FOLLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>b, c</td>
<td>$</td>
</tr>
<tr>
<td>A</td>
<td>a, B</td>
<td>a, b, c</td>
</tr>
<tr>
<td>B</td>
<td>B, C</td>
<td>a, b, c, $</td>
</tr>
</tbody>
</table>

(b) Construct the transition diagram of the DFA that models the stack. Number the states, show marked productions, and identify final states by two circles.
(c) Construct the LR parse table that corresponds to the transition diagram drawn in part b. (Note: all the rows and columns given may not be needed. **If there are multiple items for an entry, write all in the entry.**)

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>$</th>
<th>S</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s3</td>
<td>s4</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>acc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>r3</td>
<td>r3</td>
<td>r3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>r4</td>
<td>r4</td>
<td>r4</td>
<td>r4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>r5</td>
<td>r5</td>
<td>r5</td>
<td>r5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>s7</td>
<td>s3</td>
<td>s4</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>r1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>r2</td>
<td>r2</td>
<td>r2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8. (8 pts) Consider the following L-system.

Axiom: X g
X → g [- g] X f + Y
Y → g

angle 90
color black
lineWidth 2
distance 10

Recall that g is for drawing a line, f is for moving forward, + means change the direction by the angle clockwise, - means change the direction by the angle counterclockwise and [ ] are used for stacking operations.
Assume a g drawn with distance 10 and lineWidth 2 is about this size |

a. Render the L-system and draw the axiom if there is a visual picture for it.

\[
\text{g [- g] X f + Y g}
\]

b. Give the first string in the language (after the axiom) and draw it.

\[
g [- g] g [- g] X f + Y f + g g
\]

b. Give the second string in the language (after the axiom) and draw it.
9. (6 pts) **Pumping Lemma for CFL's** Let $L$ be any infinite CFL. Then there is a constant $m$ depending only on $L$, such that for every string $w$ in $L$, with $|w| \geq m$, we may partition $w = uvxyz$ such that:

- $|vxy| \leq m$, (limit on size of substring)
- $|vy| \geq 1$, ($v$ and $y$ not both empty)
- For all $i \geq 0$, $uv^i xy^i z \in L$

Consider $L = \{a^{2n} b^p c^n \mid p > n, n > 0\}$ $\Sigma = \{a, b, c\}$.

Prove $L$ is not a context-free language.

You only have to fill in the parts below. Assume $L$ is a context-free language.

(a) Choose $w = a^{2m} b^{m+1} c^m$

(b) Prove the case when $v = a^{t_1}$ and $y = b^{t_2}$

- $i = 0$ $uv^i xy^i z = u x z = a^{2m-t_1} b^{m+1-t_2} c^m \notin L$
- $t_1 = 0$ $\Rightarrow t_2 > 0$ $\Rightarrow n_a(w) \leq n_c(w)$
- $t_1 > 0$ $\Rightarrow n_a(w) \neq 2 \times n_c(w)$
- $\Rightarrow L$ is not CFL

(c) Prove the case when $v = c^{t_1}$ and $y = c^{t_3}$

- $i = 0$ $uv^i xy^i z = u x z = a^{2m} b^{m+1} c^{m-t_1-t_3} \notin L$
- $n_a(w) \neq 2 \times n_c(w)$
10. (10 pts) Construct a one-tape Turing machine (using a transition diagram) that accepts the following language:

$L = \{a^m b^n \mid m > n > 0\} \Sigma = \{a, b\}$. For example, $aabb$ would be accepted, $aaaaabb$ would be rejected (not more $b$'s than $a$'s) and $babb$ would be rejected (more $b$'s than $a$'s, but not all $a$'s first).

In drawing the transition diagram, remember to identify the start state by an arrow and final states by double circles. Format of labels are $a; b, R$ where $a$ is the symbol read on the tape, $b$ is the symbol written to the tape and $R$ is the direction moved (you can use $L$ and $R$ for directions.)

$|w| = n + m$. What is the worst case running time (big-Oh) of your TM? $\Theta(n^2 + m)$
11. (10 pts) Construct a TM (using building blocks) for the following function. $w \in \{a, b\}^*$, $|w| > 1$, $f(w) = w'$ (where $w'$ is the same as $w$ but with all the $b$'s removed except for the last $b$).

For example, $f(ababb) = aab$ and $f(baaba) = aaba$.

See the building block notation on the next page. Make sure the tape head is pointing to the leftmost symbol of the output.

Assume $|w| = n$. What is the running time in terms of $n$ (big-Oh) of your TM?

$O(n^2)$