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) $L=\{a^n b^m c^p \mid p = n \times m, n > 0, m > 0 \}, \Sigma = \{a, b, c\}$. \hspace{1cm} \underline{\hspace{2cm}}$

   (b) $L=\{a^n b^{2m} \mid m > n, n > 0\}, \Sigma = \{a, b\}$. \hspace{1cm} \underline{\hspace{2cm}}$

   (c) $L=\{b^n c^m a^{2n} \mid n > m, 100 > m > 0\}, \Sigma = \{a, b, c\}$. \hspace{1cm} \underline{\hspace{2cm}}$

   (d) $L=\{w \in \Sigma^* \mid n_a(w) \text{ is an odd number and } n_b(w) \text{ is divisible by 5} \}, \Sigma = \{a, b, c\}$. \hspace{1cm} \underline{\hspace{2cm}}$

   (e) $L=\{a^n! \mid n > 0\}, \Sigma = \{a\}$. \hspace{1cm} \underline{\hspace{2cm}}$

   (f) $L=\{w \in \Sigma^* \mid n_a(w) = n_b(w) \text{ and } n_b(w) \text{ is an even number} \}, \Sigma = \{a, b\}$. \hspace{1cm} \underline{\hspace{2cm}}$

2. (12 pts) Answer TRUE or FALSE to each of the statements below.

   (a) In converting a CFG into an NPDA that models the LL(1) parsing process, the NPDA will have three states. (TRUE or FALSE?)

   (b) In LR(1) parsing the “accept” action can appear at more than one entry in the LR(1) parse table. (TRUE or FALSE?)

   (c) A dependency graph constructed in the process of removing unit productions will never have a cycle because it is a directed graph. (TRUE or FALSE?)

   (d) There exists some CFG $G_1$ such that $G_2 = G_1^C$ and $G_2$ is also a CFG. (Note $G_2$ is the complement of $G_1$). (TRUE or FALSE?)

   (e) Suppose $G$ is a CFG with no left-recursive productions, then $G$ is an LL(1) grammar. (TRUE or FALSE?)
(f) Suppose M1 is a TM that halts on all inputs, then there exists an NPDA M2 such that L(M1)=L(M2). (TRUE or FALSE?)

3. (3 pts) Let M be a TM, M=(Q,Σ,Γ,δ,q_0,B,F). Give the formal definition of δ.

4. (4 pts) Consider the following grammar. Using this grammar, give a string and its derivation that shows both d immediately following A and e immediately following A at different parts of the derivation.

   S → bAB | c
   A → aAd | b
   B → eB | λ

5. (3 pts) Give a grammar that is in Greibech Normal Form (GNF) and is also a regular grammar.

6. (3 pts) The following grammar is LL(k) for what value of k?

   S → ABcd| aCa
   A → aA | Ba
   B → cC | e
   C → aacd

7. (4 pts) Here is the algorithm for removing λ-productions.

   **To Remove λ-productions**

   (a) Let V_n = {A | ∃ production A→λ}
   (b) Repeat until no more additions
      • if B→A_1A_2...A_m and A_i∈ V_n for all i, then put B in V_n
   (c) Construct G’ with productions P’ s.t.
• If \( A \rightarrow x_1x_2 \ldots x_m \in P, \ m \geq 1 \), then put all productions formed when \( x_j \) is replaced by \( \lambda \) (for all \( x_j \in V_n \)) s.t. \( |\text{rhs}| \geq 1 \) into \( P' \).

Show all steps in removing the \( \lambda \)-productions from the following grammar.

\[
\begin{align*}
S & \rightarrow cAB \\
A & \rightarrow SAa \mid \lambda \\
B & \rightarrow BbC \mid \lambda \\
C & \rightarrow AB \mid c
\end{align*}
\]

8. (10 pts) Consider the following grammar (DO NOT change the grammar):

\[
\begin{align*}
S & \rightarrow bAS \mid B \\
A & \rightarrow aBA \mid \lambda \\
B & \rightarrow cBd \mid Ac
\end{align*}
\]

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></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></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>d</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(c) Is this grammar an LL(1) grammar? Explain.
9. (15 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.

\[
\begin{align*}
0) & \quad S' \rightarrow S \\
1) & \quad S \rightarrow bAB \\
2) & \quad A \rightarrow AA \\
3) & \quad A \rightarrow \lambda \\
4) & \quad B \rightarrow BB \\
5) & \quad B \rightarrow b
\end{align*}
\]

(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></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></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.)
10. (8 pts) Consider the following L-system.

Axiom: $g + X + g$

$X \rightarrow Y g [ - - g ] + X$

$Y \rightarrow f g$

angle 45

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.

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

b. Give the second string in the language and draw it.

11. (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 $i \geq 0$, $uv^ixy^iz \in L$

Consider $L=\{a^n b^m c^{2n} \mid m < n, m > 0\}$ \(\Sigma = \{a, b, c\}\).

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 $w =$

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

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

12. (9 pts) Construct a one-tape TM (using a transition diagram) that accepts the following language:

$L=\{w \in \Sigma^* \mid n_a(w) > n_b(w)\}$ \(\Sigma = \{a, b\}\). For example, $ababb$ would be rejected and $aaabbbbaa$ would be accepted.

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.)

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

13. (9 pts) Construct a TM (using building blocks) for the following function. $w \in \{a, b\}^+$ $f(w) = w^R$ (the reverse of the string).

For example, $f(abbba) = bbbba$ and $f(aabbaaa) = aaaaabbaa$.

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?
Notation for Simplifying Turing Machines

Suppose $\Gamma = \{a,b,c,B\}$

$z$ is any symbol in $\Gamma$

$x$ is a specific symbol from $\Gamma$

1. $s$ - start
2. $R$ - move right
3. $L$ - move left
4. $x$ - write $x$ (and don’t move)
5. $R_a$ - move right until you see an $a$ (note that this moves right at least one square before it checks for $a$).
6. $L_a$ - move left until you see an $a$
7. $R_{a|b}$ - move right until you see an $a$ or $b$
8. $L_{a|b}$ - move left until you see an $a$ or $b$
9. $R_{\neg a}$ - move right until you see anything that is not an $a$
10. $L_{\neg a}$ - move left until you see anything that is not an $a$
11. $h$ - halt in a final state
12. $\frac{a,b}{w}$

If the current symbol is $a$ or $b$, let $w$ represent the current symbol.

13. $C$ - copy a string, $F(w) = w0w$, makes a copy of the string and inserts a 0 between the original and the copy.
14. $S_L$ - Shift the string that is to the right of the tape head (up to a blank), to the left, writing over the symbol the tape head is pointing to.
15. $S_R$ - Shift the string that is to the left of the tape head (up to a blank), to the right, writing over the symbol the tape head is pointing to.