1. Consider the following two DFAs, $A_1$, $A_2$, using the technique of cross-product of two machines, create a new one that accepts the intersection of $L(A_1)$ and $L(A_2)$. Be sure to identify the state and final states in your composite DFA. I recommend you use a table here to avoid crossing arcs.

![DFA Diagrams]

<table>
<thead>
<tr>
<th>State/Input</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start AD</td>
<td>BD</td>
<td>AE</td>
</tr>
<tr>
<td>AE</td>
<td>BE</td>
<td>AD</td>
</tr>
<tr>
<td>BD</td>
<td>CD</td>
<td>BE</td>
</tr>
<tr>
<td>Final BE</td>
<td>CE</td>
<td>BD</td>
</tr>
<tr>
<td>CD</td>
<td>AD</td>
<td>BE</td>
</tr>
<tr>
<td>Final CE</td>
<td>AE</td>
<td>BD</td>
</tr>
</tbody>
</table>

2. Let $A = (\{q_1, \ldots, q_4\}, \{0,1\}, q_1, \{q_2\})$ be some DFA. Assume you have computed the sets, $R_{i,j}^k$, for $0 \leq k \leq 3$, $1 \leq i \leq 4$, $1 \leq j \leq 4$. How do you compute

$L(A) = R_{1,2}^4$: based on the previously computed values of the $R_{i,j}^k$’s?

$R_{1,2}^4 = R_{1,2}^3 + R_{1,4}^3 \cdot (R_{4,4}^3)^* \cdot R_{4,2}^3$

3. Let $L_1$, $L_2$ be Non-Regular CFLs; $R$ be Regular; Answer is about $S$ and there can be more than one cell per row that has an $X$.

<table>
<thead>
<tr>
<th>Definition of $S$ / Characterization of $S$</th>
<th>Can be Regular</th>
<th>Can be a non-Regular CFL</th>
<th>Can be more complex than a CFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S = L_1 - L_2$, where $-$ is set difference</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td>$S = L_1 \cdot R$, where $\cdot$ is concatenation</td>
<td>$X$</td>
<td>$X$</td>
<td></td>
</tr>
<tr>
<td>$S = \sigma(R)$, where $\sigma$ is a homomorphism</td>
<td>$X$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S = \sigma(L_1)$, where $\sigma$ is a homomorphism</td>
<td>$X$</td>
<td>$X$</td>
<td></td>
</tr>
</tbody>
</table>
6 4.   Let \( L \) be defined as the language accepted by the following DFA \( A_3 \).

\[
\begin{array}{c}
\text{A3} \\
\text{A} & \xrightarrow{0} & \text{B} & \xrightarrow{1} & \text{C} \\
\text{1} & \xrightarrow{1} & \text{B} & \xrightarrow{1} & \text{C}
\end{array}
\]

Present the regular equations associated with each of \( A_3 \)'s states, solving for the regular expression associated with the language recognized by \( A_3 \). You must finish by showing the final expression for the language accepted by this automaton.

\[
\begin{align*}
A &= \lambda + C0 + A1 = (\lambda + C0)1* = 1* + C01* = 1* + B001* \\
B &= A0 + C1 + B1 = 1*0 + B001*0 + B01 + B1 = 1*0 (001*0 + 01 + 1)* \\
C &= B0 \\
L &= B = 1*0 (001*0 + 01 + 1)*
\end{align*}
\]

4 5.   Looking back at \( A_3 \), above, write a Right Linear Grammar that generates the language accepted by \( A_3 \). Note: You must fill in the list of non-terminals and specify the rules, \( R \).

\[
G = ( \{ A, B, C \}, \{ 0, 1 \}, R, A)
\]

\[
\begin{align*}
A &\rightarrow 1A | 0B \\
B &\rightarrow 1B | 0C | \lambda \\
C &\rightarrow 0A | 1B
\end{align*}
\]

5 6.   Analyze the language, \( L = \{ a^i b^j \mid i < j, j > 0 \} \), proving it is non-regular by showing that there are an infinite number of equivalence classes formed by the relation \( R_L \) defined by:

\[ x R_L y \text{ if and only if } \forall z \in \{a,b\}^*, xz \in L \text{ exactly when } yz \in L . \]

You don’t have to present all equivalence classes, but you must demonstrate a pattern that gives rise to an infinite number of classes, along with evidence that these classes are distinct.

Consider the right invariant equivalence classes of \( R_L \), \( [a^i]_{R_L} \), where \( i \geq 0 \).

Clearly, \( a^{i+j}b^{i+j} \) is in \( L \)

However, \( a^ib^{i+1} \) is not in \( L \) whenever \( j > i \)

Thus, \( [a^i]_{R_L} = [a^i]_{R_L} \text{ iff } i = j \)

This gives rise to a separate equivalence class for each \( i \geq 0 \) and so \( R_L \) has infinite index and so, by the Myhill-Nerode Theorem, \( L \) cannot be Regular.
6 7. Let $L$ be defined as the language accepted by the NFA $A_4$:

Using the technique of replacing transition letters by regular expressions and then ripping states from a GNFA to create new expressions, develop the regular expression associated with the automaton $A_4$ that generates $L$. I have included the states of a GNFA associated with removing states $A$, $C$ and then $B$, in that order. You must use this approach of collapsing one state at a time, showing the resulting transitions with non-empty regular expressions.

3 8. Define $\text{WeirdSub} (L) = \{ uv | u,v \in \Sigma^+ \text{ and } (\exists x,y,z \in \Sigma^+) \text{ where } xuyvz \in L \}$

Show CFLs are closed under $\text{WeirdSub}$. You should find it useful to employ the substitution $f(a) = \{a, a'\}$, and the homomorphisms $g(a) = a'$ and $h(a) = a$, $h(a') = \lambda$. Here $a \in \Sigma$ and $a'$ is a new symbol associated with $a$. You do not need to show your construction works, but it must be based on the meta technique I showed in class.

$$\text{WeirdSub}(L) = h (f(L) \cap g(\Sigma^+) \Sigma^+ g(\Sigma^+) \Sigma^+ g(\Sigma^+))$$
9. Given a DFA denoted by the transition table shown below, and assuming that 1 is the start state and 1, 2, and 4 are final states, fill in the equivalent states matrix I have provided. Use this to create an equivalent, minimal state DFA.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Show minimal state equivalent automaton below. Mark the start state and final state(s). Note: if 1, 2 and 3 are indistinguishable, label the merged state as 123.

6. Use the Pumping Lemma for CFLs to show that the following language $L$ is not Context Free.
$L = \{ a^i b^j c^k \mid i < \min(j,k), j,k > 0 \}.$
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 four steps for you and even started the final step by recommending you can split this into two mutually exclusive cases.

**ME: Assume $L$ is Context Free**

**PL: Provides a whole number $N>0$ that is the value associated with $L$ based on the Pumping Lemma**

**ME: I chose $a^N b^{N+1} c^{N+1}$ which clearly belongs to $L$ and has length $\geq N.$**

**PL: Breaks $a^N b^{N+1} c^{N+1}$ into five parts $uvwxy,$ where $|vwx|\leq N$ and $|vx|>0.$ Also, the PL states that $uv^iwx^iy$ is in $L$ for all $i\geq 0.$**

**Me: Split this into two cases:**

**Case 1: vx contains at least one a. Set $i=2,$ then there are now at least $N+1$ $a$’s and at least $N+1$ $b$’s, and, since $|vwx|\leq N,$ $vx$ cannot span both $a$’s and $c$’s, and so there are still $N+1$ $c$’s. Thus, there are now at least as many $a$’s as $c$’s and hence as the minimum of the $b$’s and $c$’s and so $uv^2wx^2y$ is not in $L.$**

**Case 2: vx contains no a’s. Set $i=0,$ then there are now either fewer than $N+1$ $b$’s or fewer than $N+1$ $c$’s or fewer than $N+1$ of both. However, there are still $N$ a’s. Thus, there are now at least as many a’s as the minimum of the $b$’s and $c$’s and hence $uv^0wx^0y = uy$ is not in $L.$**

The above cover all cases and so $L$ is not a CFL.
11. Demonstrate the steps associated with a Reduced, then Chomsky Normal Form grammar.

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

$$
S \rightarrow AB1 | BA0 \\
A \rightarrow 0A0 | B0 \\
B \rightarrow 1B1 | \lambda
$$

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

Nullable = \{ B \}

$$
S \rightarrow AB1 | BA0 | A1 | A0 \\
A \rightarrow 0A0 | B0 | 0 \\
B \rightarrow 1B1 | 11
$$

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

$$
S \rightarrow AB1 | A \\
A \rightarrow 0A0 | B \\
B \rightarrow 1B1 | 0
$$

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

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

$$
S \rightarrow AB1 | 0A0 | 1B1 | 0 \\
A \rightarrow 0A0 | 1B1 | 0 \\
B \rightarrow 1B1 | 0
$$

4 c.) Consider the reduced context-free grammar $G_3 = (\{ S, A, B, W, X \}, \{ 0, 1 \}, R_3, S)$, where $R_3$ is:

$$
S \rightarrow ABBA | BAAB \\
A \rightarrow 0W | 0 \\
B \rightarrow 1X | 1 \\
W \rightarrow A1 \\
X \rightarrow B0
$$

Convert to an equivalent Chomsky Normal Form grammar $G_3'$. Show all rules.

$$
S \rightarrow <AB><BA> | <BA><AB> \\
A \rightarrow <0>W | 0 \\
B \rightarrow <1>X | 1 \\
W \rightarrow A<1> \\
X \rightarrow B<0> \\
<AB> \rightarrow AB \\
<BA> \rightarrow BA \\
<0> \rightarrow 0 \\
<I> \rightarrow I
$$
6 12. Present the CKY recognition matrix for the string \textbf{abaab} assuming the Chomsky Normal Form grammar, $G = ( \{ S, A, B, X, Y, Z \}, \{ a, b \}, R, S )$, specified by the rules $R$.
Note: This matrix is densely populated.

\begin{align*}
S & \rightarrow AB \mid BA \\
A & \rightarrow XY \mid a \\
B & \rightarrow XZ \mid b \\
Y & \rightarrow AX \\
Z & \rightarrow BX \\
X & \rightarrow a \mid b
\end{align*}

\begin{tabular}{|c|c|c|c|c|}
\hline
   & a & b & a & b \\
\hline
1 & AX & BX & AX & AX & BX \\
2 & SY & SZ & Y & SY \\
3 & B & A & A & \\
4 & SZ & SY & \\
5 & A & \\
\hline
\end{tabular}

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>None</td>
</tr>
<tr>
<td>A</td>
<td>S \rightarrow AB; Y \rightarrow AX</td>
<td>S \rightarrow BA</td>
</tr>
<tr>
<td>B</td>
<td>S \rightarrow BA; Z \rightarrow BX</td>
<td>S \rightarrow AB</td>
</tr>
<tr>
<td>X</td>
<td>A \rightarrow XY; B \rightarrow XZ</td>
<td>Y \rightarrow AX; Z \rightarrow BX</td>
</tr>
<tr>
<td>Y</td>
<td>None</td>
<td>A \rightarrow XY</td>
</tr>
<tr>
<td>Z</td>
<td>None</td>
<td>B \rightarrow XZ</td>
</tr>
</tbody>
</table>

Is \textbf{abaabb} in $L(G)$? \textbf{N}