1. Give a regular expression for each state $q$ that specifies the set of words that will take you from the start state to state $q$.

\[
L_1 = (a^*b + b^*a)^* \\
L_2 = L_1 \cdot a \\
L_3 = L_1 \cdot b \\
L_4 = L_1 \cdot (a + b) (a + b)^* 
\]

2. Minimize the following FA. Show your work. Show the FA in case the number of states was reduced.
3. Give an NFA that accepts all strings over \( \{0, 1\} \) containing at least three 1's.

\[
(0+1)^* 1 (0+1)^* 1 (0+1)^* 1 (0+1)^*
\]

4. Use the "subset construction method" to convert the following NFA to an FA. Label the NFA you produce with the subsets of states of original NFA.
5. Convert the following \( \lambda \text{NFA} \) into a regular expression. Use the algorithm given in class that eliminates one node at a time and creates arcs that are labeled with regular expressions. First eliminate state 2.

\[
(\alpha b^* a + a b^* a (b + a b^* a)^* (a b^* a + \lambda))
\]
6. Give a concise description of the language over the alphabet \{0, 1\} generated by the following grammar.

\[
\begin{align*}
S & \rightarrow A \mid B \\
B & \rightarrow BO \mid Q \\
A & \rightarrow 1A \mid Q \\
Q & \rightarrow 1Q0 \mid \lambda \\
A & \Rightarrow 1^* Q \\
B & \Rightarrow Q 0^* \\
Q & \Rightarrow \{1^m0^n : n \geq 0\} \\
S & \Rightarrow 1^* \{1^m0^n : n \geq 0\} \cup \{1^m0^n : n \geq 0\} 0^* \\
L(G) & = 1^* 0^*
\end{align*}
\]

Any FA accepting...

7. Show that the following regular language must have at least three states: \((a + b)^*aba.\)

\[
\begin{align*}
\Lambda, a, ab & \text{ p.c.l.} \\
\text{witness } \epsilon & \\
\Lambda, a & \Rightarrow a \\
cr, ab & \Rightarrow ba \\
\Lambda, ab & \Rightarrow a \\
\Lambda & \notin L \text{ a.e. } \Lambda & \in L \text{ a.e. } \\
\Lambda & \notin L \text{ a.e. } \Lambda & \notin L \text{ a.e. } ab \notin L
\end{align*}
\]

In each case one \(Z\) extension lies \(\notin L\) and the other not!
8. Show that if a $L$ over some finite alphabet $\Sigma$ is regular then the language $\bar{L}$ of all subwords of words in $L$ is also regular.

Formally $\bar{L}$ is defined as $\{v \in \Sigma^* | \exists u, w \in \Sigma^* \text{ such that } uvw \in L\}$.

Either give a construction or use the closure properties shown in class. "Briefly" justify your answer.

I) $Subw(L) = Prefix(Suffix(L))$

$= \{ z : \exists u \in \Sigma^* \text{ s.t. } uz \in L \}$

$= \{ v : \exists w \in \Sigma^* \text{ s.t. } vw \in L' \}$

$= \{ u : \exists w \in \Sigma^* \text{ s.t. } uuw \in L \}$

II) Let $M$ be minimized FA accepting $L$.

Remove trap states (if there is one).

The resulting NFA $M'$ (might have no states) has the following property (since $M$ was minimized):

Any state is reachable from start state and any state can reach the final state.

$M$: 

\[ \begin{array}{c}
\text{new start state} \\
\text{new final state}
\end{array} \]

$M'$: 

\[ \begin{array}{c}
\text{new start state} \\
\text{new final state}
\end{array} \]
\( L(\tilde{M}) = \tilde{c} \)

**Proof:** Assume \( u \in L(\tilde{M}) \). Then \( u \) labels a path between the states \( p \) and \( q \). Let \( u \) be a word leading you from the start state to \( p \). Let \( w \) be a word leading from \( q \) to a final state.

Since \( u v w \in L \), \( v \) is a subword of \( u \) and \( w \). Thus \( u \in \tilde{C} \).

Assume \( v \in \tilde{C} \). Then there exists \( u \) such that \( u v w \in L \).

\( u \) labels a path in \( M' \).

\[ M' \]

\[ u \rightarrow (p) \rightarrow v \rightarrow q \]

The \( \lambda \) transitions \( u \) pick out the subpath corresponding to \( v \). Thus \( v \in L(\tilde{M}) \).
9. Use the Pumping Lemma for regular languages to show that

\[ \{(ab)^n c^n \mid n \geq 0\} \]

is not regular.

You can use the following Pumping Lemma:

For every regular language \( L \) there is a constant \( N \) such that each word \( x \in L \) of length at least \( N \) can be written as \( uvw \) such that the following holds:

1. \( |uv| \leq N \),
2. \( v \) is not the empty word and
3. for all \( i \geq 0 \), \( uv^iw \in L \).

Assume \( L \) is reg.

Let \( N \) be const. of PL

Choose \( x = (ab)^N c^N \)

Since \( |x| \geq N \), i)-iii) hold.

By i) \( uv \in (a+b)^* \)

By ii) \( v \in (a+b)^* \)

\( \text{w.l.o.g. } v \) contains an "a"

\( uw \) contains \( N "c" \)'s and less than \( N "a" \)'s

Thus \( uv^0w = uvw \notin L \)

This contradicts iii)