NOTE: \( n_a(w) \) means the number of a’s in the string \( w \).

1. (12 pts) Complete or answer the following.

\[
L_1 = \{a, \{b, c\}\}, \Sigma = \{a, b, c\} \\
L_2 = \{a, bc\}, \Sigma = \{a, b, c\} \\
L_3 = \{w \in \Sigma^* \mid n_a(w) > n_b(w) + n_c(w)\}, \Sigma = \{a, b, c\} \\
L_4 = b^*a^*c^*, \Sigma = \{a, b, c\}
\]

(a) \( |L_1| = \)  
(b) \( 2^{L_2} = \)  
(c) \( L_2 \circ L_2 = \)  
(d) \( L_3 \cap L_4 = \)  
(e) \( L_2 \times L_2 = \)  
(f) \( L_3 = \)  

2. (22 pts) Answer TRUE or FALSE to each of the statements below.

(a) If \( M \) is an NPDA and \( M \)'s transition diagram does not have a cycle, then there exists an NFA \( M' \) such that \( L(M) = L(M') \). (TRUE or FALSE?)

(b) There exists an NFA that recognizes valid arithmetic expressions formed over \( \Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, (, )\} \). (TRUE or FALSE?)

(c) When converting an NFA \( M \) to a DFA \( M' \) using the algorithm we discussed in class, the number of states in \( M' \) ≥ the number of states in \( M \). (TRUE or FALSE?)

(d) Consider the following statement involving regular expressions.
\[ a^*(ab + b) = aa^*b \] (TRUE or FALSE?)

(e) The following grammar \( G \) is a regular grammar. (TRUE or FALSE?)
\[
S \to aS \mid bbB \mid c \\
B \to bbS \mid \lambda
\]

(f) If \( R \) is a regular expression, then there exists an NPDA \( M \) with \( \Gamma = \{Z\} \) such that \( L(R) = L(M) \). (TRUE or FALSE?)

(g) If \( G_1 \) and \( G_2 \) are regular grammars, then there exist a DFA \( M \) such that \( L(M) = L(G_1) \cap L(G_2) \). (TRUE or FALSE?)

(h) If \( M \) is an NPDA, then there exists a DCFG \( G \) such that \( L(M) = L(G) \). (TRUE or FALSE?)
(i) \( L = \{ w \in \Sigma^* \mid n_a(w) > 2 \cdot n_b(w) \} \), \( \Sigma = \{a, b\} \). \( L \) is regular. (TRUE or FALSE?)

(j) \( L = \{ a^n b^p c^q \mid 0 < n < q, q < 100, p > n \} \), \( \Sigma = \{a, b, c\} \). \( L \) is regular. (TRUE or FALSE?)

(k) \( L = \{ w \in \Sigma^* \mid n_a(w) \text{ is even and } n_b(w) \text{ is odd} \} \), \( \Sigma = \{a, b, c\} \). \( L \) is regular. (TRUE or FALSE?)

3. (3 pts) With respect to a DFA, which of the following could be infinite? (circle all that apply).
   a) \( Q \)
   b) \( \Sigma \)
   c) the input tape
   d) \( \delta \)
   e) the language of strings accepted by \( M \)

4. (3 pts) Give an example of a context-free grammar \( G \) that is not a regular grammar, but \( L(G) \) is regular.

5. (4 pts) Consider the following grammar.
   \[ S \rightarrow SS \mid aSa \mid b \mid \lambda \]
   A) Give a right-most derivation for the string \( babba \).
   B) Give a parse tree for the string \( babba \)

6. (8 pts) Draw a DFA for the following language. You do not have to show trap states (show the transition diagram, indicate the start state by a short arrow, and final states by double circles.)
   \( L = \{ w \in \Sigma^* \mid n_b(w) \text{ is a multiple of 3 and } abb \text{ is a substring of } w \} \), \( \Sigma = \{a, b\} \).
   For example, \( babbaa, aabb \) and \( bbbabb \) are in \( L \).

7. (6 pts) Consider the following DFA. Note there are two arcs from \( q_0 \) to \( q_3 \), one labeled \( a \) and one labeled \( b \). It is shown as one arc with two labels.
   a) Show states \( q_0 \) and \( q_2 \) are distinguishable with an appropriate string. Explain.
   b) Give the states in the minimal state DFA (you do not need to show the arcs). Each state should indicate which states it represents from the original DFA. For example you could list one state as \( 0, 1, 2 \) if states 0, 1 and 2 in the original DFA can be combined to form a state in the minimal state DFA.
8. (10 pts) Consider \( L = \{ a^n b^m c^p \mid n < m+p \text{ and } n \text{ is even, } p > 0, n \geq 0, m > 0 \} \). Draw the transition diagram for a nondeterministic pushdown automaton \( M \) that accepts \( L \) by final state. (Remember to identify the start state by an arrow and final states by double circles. Format of labels are \( a, b; cd \) where \( a \) is the symbol on the tape, \( b \) is the symbol on top of the stack that is popped, and \( cd \) are pushed onto the stack (with \( c \) on top of \( d \)). \( Z \) is on top of the stack when \( M \) starts. ).

(a) First list 3 strings in \( L \).

(b) Now draw the transition diagram.

9. (6 pts) **Pumping Lemma:** Let \( L \) be an infinite regular language. \( \exists \) a constant \( m > 0 \) such that any \( w \in L \) with \( |w| \geq m \) can be decomposed into three parts as \( w = xyz \) with

\[
\begin{align*}
|xy| & \leq m \\
|y| & \geq 1 \\
xy^iz & \in L \quad \text{for all } i \geq 0
\end{align*}
\]

**Use the Pumping Lemma to prove** the language \( L \) below is not regular.

\( L = \{ a^n b^p c^q \mid p < n + q, p > 0, n > 0, q > 0 \} \), \( \Sigma = \{ a, b, c \} \).

For example, \( aabcc \) and \( abc \in L \).
Proof: (SHOW ALL STEPS! Some have been started for you.)

Assume ______________________________________________________________________

Choose w = ______________________________________________________________________

Show there is no way to partition this string w=xyz such that the properties of the
pumping lemma hold.

10. (8 pts) Consider the following property, ReplaceSomeEvenAWithB. If L is a regular
language, then ReplaceSomeEvenAWithB(L)= strings from L that have some even a
(the 2cd a, 4th a, 6th a, etc) replaced with b. If there is a string w that does not have
at least two a’s in the string, then w is NOT in ReplaceSomeEvenAWithB.

For example, if caabb is in L, then cabbb is in ReplaceSomeEvenAWithB(L), the second
a was replaced. If aaaaaaab is in L, then aaabaaab is in ReplaceSomeEvenAWithB(L),
the 4th a was replaced. Note that the 2cd or 6th a could have been replaced instead.
If aba is in L, then abb is in ReplaceSomeEvenAWithB(L).

If ab is in L, then ab does not generate a string in ReplaceSomeEvenAWithB(L).

Prove that ReplaceSomeEvenAWithB(L) is a regular language.