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

1. (14 pts) Complete or answer the following.

$L_1 = \{b, c\}, \Sigma = \{b, c\}$
$L_2 = \{a, ba, c\}, \Sigma = \{a, b, c\}$
$L_3 = (ba)^*(b)^*, \Sigma = \{a, b\}$
$L_4 = \{w \in \Sigma^* | n_b(w) = 2 \cdot n_a(w)\}, \Sigma = \{a, b\}$

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

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

(a) If $G$ is a regular grammar, then there exists a CFG $G'$ such that $L(G) = L(G')$. (TRUE or FALSE?)

(b) Given an NPDA $M$ with only one symbol in the alphabet, then there exists an NFA $M'$ such that $L(M) = L(M')$. (TRUE or FALSE?)

(c) If $M$ is an NFA with more than one final state, then $L(M)$ is an infinite language. (TRUE or FALSE?)

(d) The following grammar $G$ is a regular grammar. (TRUE or FALSE?)

\[
\begin{align*}
S & \rightarrow a \mid B \\
B & \rightarrow Bb \mid \lambda
\end{align*}
\]

(e) Consider the following statement involving regular expressions.

\[(a + \lambda)^* \emptyset^*(b + a) = a^*b + a\] (TRUE or FALSE?)

(f) Given an NPDA $M$ such that no symbols are ever pushed onto the stack, then there exists a regular expression $R$ such that $L(M) = L(R)$. (TRUE or FALSE?)

(g) If $L$ has exactly 5 million strings in the language, then there is a DFA $M$ such that $L = L(M)$. (TRUE or FALSE?)

(h) $L = \{w \in \Sigma^* | n_a(w) > n_b(w) \text{ and there are an even number of } a\text{'s} \}, \Sigma = \{a, b\}$. $L$ is regular. (TRUE or FALSE?)
(i) \(L = \{a^n a^n b^m \mid n > 0, m > 0\}, \Sigma = \{a, b\}. \) L is regular. (TRUE or FALSE?)

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

(k) \(L = \{w \in \Sigma^* \mid n_a(w) > 100 \text{ and } n_b(w) \text{ is an odd number} \} \Sigma = \{a, b\}. \) L is regular. (TRUE or FALSE?)

3. (4 pts) Consider the following grammar with start variable S.

\[
S \rightarrow aSBb \mid aBB \mid a \\
B \rightarrow bBb \mid \lambda
\]

A) Give a right-most derivation for the string \(aabbb.\)

B) Give two different parse trees for the string \(aabbb.\)

4. (5 pts) Write a CFG G for the following language:

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

For example, \(aabccc \in L.\)

5. (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.)

\(L = \{w \in \Sigma^* \mid n_a(w) \text{ is not divisible by 3 and } w \text{ has the substring } baa\}, \Sigma = \{a, b\}.\)

For example, \(baab, baabaa, aabaabbbb, \text{ and } baaabbbb \) are in \(L.\)

6. (6 pts) Consider the following DFA. Note there are two arcs from \(q_0\) to \(q_2,\) one labeled \(a\) and one labeled \(b.\) It is shown as one arc with two labels.

(a) Show states \(q_0\) and \(q_3\) 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.

7. (10 pts) Consider \(L = \{a^n b^p c^m \mid m > 2n + p, n > 0, p > 0\}, \Sigma = \{a, b, c\}.\) 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.

8. (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

$$\begin{align*}
|xy| &\leq m \\
|y| &\geq 1 \\
x y^i z &\in L \text{ for all } i \geq 0
\end{align*}$$

**Use the Pumping Lemma to prove** the language $L$ below is not regular.

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

**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.

9. (8 pts) Consider the following property, ReplaceApastB. If $L$ is a regular language, then $\text{ReplaceApastB}(L) =$ strings from $L$ that replace the first $a$ that is past the first $b$, with a $c$. If there is a string $w$ in $L$ that does not have an $a$ past a $b$, then that does not put a string in $\text{ReplaceApastB}(L)$.

For example, if $aabbaabaa$ is in $L$, then $aabbcabaa$ is in $\text{ReplaceApastB}(L)$. (The first $b$ is the third character in the string and the first $a$ past that $b$ is the fifth character in the string.) If $aba$ is in $L$, then $abca$ is in $\text{ReplaceApastB}(L)$. If $abccaa$ is in $L$, then
abc is in ReplaceApastB(L)). If aaabb is in L, then aaabb does not generate a string in ReplaceApastB(L).

Start the proof to show that ReplaceApastB(L) is a regular language. You need to explain how to convert a DFA M for L into a DFA or NFA that represents the language ReplaceApastB(L).