1. Honor code (2 pts)
   Print your name to acknowledge the Duke Community Standard

2. (18 pts) Complete or answer the following.

   \[ L_1 = \{ba, cb\}, \Sigma = \{a, b, c\} \]
   \[ L_2 = \{aba, cb, cc\}, \Sigma = \{a, b, c\} \]
   \[ L_3 = b(ba)^*b, \Sigma = \{a, b\} \]
   \[ L_4 = (a + b)(a + b)^*, \Sigma = \{a, b\} \]
   \[ L_5 = \{w \in \Sigma^* \mid n_b(w) > n_a(w) + 1\}, \Sigma = \{a, b\} \]

   Type your answers into the boxes. For emptyset use \{\}. For empty string use &.

   (a) \[ |L_1 \cup L_2| = \]

   (b) \[ L_1 \circ \{ba\} = \]

   (c) \[ L_1 \cap L_2 = \]

   (d) \[ L_3 \cap L_4 = \]

   (e) \[ L_3 \cap L_5 = \]

   (f) \[ L_1 - L_2 = \]

   (g) \[ |L_1 \circ L_2 \circ L_2| = \]

   (h) \[ L_1 \times L_1 = \]

   (i) \[ |2^{L_2}| = \]
3. (16 pts) Answer TRUE or FALSE to each of the statements below.

(a) Any NFA $M_1$ can be converted into a DFA $M_2$ such that $M_2$ has at most twice as many states as $M_1$. (TRUE or FALSE?)

(b) Suppose DFA $M$ is such that $L(M)$ is an infinite language. Then $M$ could have an infinite number of states. (TRUE or FALSE?)

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

\[
S \rightarrow Sa | Bb | \lambda \\
B \rightarrow Bbb | S
\]

(d) $\lambda \in \{ba, cb\}$

(TRUE or FALSE?)

(e) $c(a + \lambda)^*b = ca^*b$

(TRUE or FALSE?)

(f) $(a + b^*a)(ba) = aba + b^*aba$

(TRUE or FALSE?)

(g) A regular grammar with more than 10 rules must be an infinite language.

(TRUE or FALSE?)

(h) Suppose $L$ is an infinite regular language and the regular grammar for $L$ has only two variables. Then there must be at least three rules in the grammar.

(TRUE or FALSE?)
4. (8 pts) Consider the following grammar G with start variable S.

\[
S \rightarrow aaS \mid bB \mid \lambda \\
B \rightarrow bbB \mid bbS
\]

A) Give a derivation for the string \textit{aabbbbb}.

B) Give a string of length greater than 6 that is not in L(G).

C) Give a regular expression for L(G).
Note: The regular expression \textit{a*b} you would enter as: \textit{a * b}.
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 odd and every } b \text{ must be adjacent to another } b\}, \Sigma = \{a, b\}.
For example, babb, bbbaaa, aababbbb, and aaabbbbaa are in \(L\).
6. (8 pts) Consider the following DFA.

a) Give one string that is accepted by this DFA that is of length 4 or greater.

b) Consider converting this DFA into a minimal state DFA. Draw the tree that shows how states are determined to be distinguishable. The root should be labeled: 0 1 2 3 4 5 6 7, to represent all the states from q0 to q7. Beside or under each node, identify the terminal that splits the node into children, if that node has children, or other reason why you are splitting the node.

c) In the algorithm to convert a DFA into a minimal state DFA, in general is it possible
for a terminal to break a tree node named N into two or more child nodes, and then the same terminal be used again in one of N’s child nodes to break that node into two or more child nodes? Explain.
7. (8 pts) Consider the following DFA.

Note there are two arcs from \( q_0 \) to \( q_1 \), one labeled \( a \) and one labeled \( b \). It is shown as one arc with two labels.

a) Give an equivalent regular expression for this DFA.

b) Give an equivalent regular grammar for this DFA.
8. (8 pts) Consider the following property, Replace3rdBwithBB. If $L$ is a regular language, then $\text{Replace3rdBwithBB}(L) = \text{strings from } L \text{ that replace the third } b \text{ in the string with } bb$. If there is a string $w$ in $L$ that does not have at least three $b$’s, then that does not put a string in $\text{Replace3rdBwithBB}(L)$.

For example, if $abbaaba$ is in $L$, then $abbaabba$ is in $\text{Replace3rdBwithBB}(L)$. (The third $b$ is replaced with $bb$). If $abababab$ is in $L$, then $abababbbab$ is in $\text{Replace3rdBwithBB}(L)$. If $abbbba$ is in $L$, then $abbbbbbba$ is in $\text{Replace3rdBwithBB}(L)$. If $aaabb$ is in $L$, then $aaabb$ does not generate a string in $\text{Replace3rdBwithBB}(L)$, because it only has two $b$’s.

For this problem, you only need to explain how to convert a DFA $M$ for $L$ into a DFA or NFA that represents the language $\text{Replace3rdBwithBB}(L)$. That is you need to describe all the additions (copy of $M$?, additional states, additional arcs and their labels, etc.) and subtractions (removed states, removed arcs, etc) to create the new DFA or NFA that represents $\text{Replace3rdBwithBB}(L)$. You DO NOT need to prove that this construction works.