The test has eight problems. You’ll be graded on your best six answers.

1. Let $A = \{a, b, c\}$. Present the smallest binary relation on $A$ that is symmetric, transitive, and non-reflexive, and whose domain includes all of $A$. Equivalently, show a directed graph $G$ satisfying these properties:
   (1) it has three nodes (labeled $a$, $b$, and $c$),
   (2) every node has outdegree one or more,
   (3) at least one node fails to have an outgoing edge to itself,
   (4) if $(x, y)$ is an edge, then $(y, x)$ must also be an edge,
   (5) if $(x, y)$ and $(y, z)$ are edges, then $(x, z)$ must also be an edge, and
   (6) no graph satisfying properties (1) through (5) has fewer edges than $G$.

2. Prove by mathematical induction that, for all $n \geq 0$,
   \[
   \sum_{0 \leq i < n} 2^i = 2^n - 1
   \]

3. Present a context-free grammar that generates the language $L = \{xcy : x, y \in \{a, b\}^* \land n_a(x) = n_b(y)\}$
   That is, $L$ contains strings with one occurrence of $c$ in which the number of occurrences of $a$ in the prefix preceding the $c$ is equal to the number of occurrences of $b$ in the suffix following the $c$. The number of occurrences of $b$ in the prefix preceding $c$ is irrelevant, and similarly for the number of $a$’s in the suffix following the $c$. Thus, for example, $aabbacabaaabaaaa$ is a member of $L$, as there are three occurrences of $a$ before the $c$ and three occurrences of $b$ after the $c$.

   **Hint:** Introduce a variable $A$ that can generate any string in $b^*ab^*$ and a variable $B$ that can generate any string in $a^*ba^*$.

4. Consider the context-free grammar $G = (\{S, E\}, \{b\}, S, P)$, where $P$ is this set of productions:
   \[
   S \rightarrow bSE \mid b \\
   E \rightarrow bE \mid b
   \]
   (a) Describe $L(G)$ in a precise way.
   (b) Show that $G$ is an ambiguous grammar.
   (c) Show an unambiguous grammar $G'$ such that $L(G') = L(G)$. 

1
5. Show a DFA that accepts the strings over \{a, b, c\} in which neither aab nor bba occurs as a substring.

6. Present a six-state NFA that accepts precisely those strings over \{a, b\} of length three or more whose suffix of length three has at least two occurrences of a.

7. Use the subset construction to convert the following NFA into an equivalent DFA.

8. Present a regular expression that describes the language accepted by the NFA in the previous problem.

**Hint:** A good solution would be \((r + s) \cdot t\), where \(r\), \(s\), and \(t\) are regular expressions such that

- \(L(r)\) is the set of strings labeling walks that go from state 1 to state 4 but pass through neither state 2 nor 4,
- \(L(s)\) is the set of strings labeling walks that go from state 1 to state 4 but pass through neither state 3 nor 4, and
- \(L(t)\) is the set of strings labeling walks that go from state 4 to itself.