1. (10%) Let $B$ be the set of finite, ordered, unlabeled binary trees. Prove by induction on $B$ that every finite ordered binary tree has one more leaf than internal nodes. That is, if $\ell(t)$ is the number of leaves in the tree $t$ and $i(t)$ the number of internal nodes, then $\ell(t) = i(t) + 1$.

**Solution.** Base: When $t$ is the singleton tree, $\ell(t) = 1 = 0 + 1 = i(t) + 1$.

Step: Suppose $t$ is obtained from $t_0$ and $t_1$. Assume (IH) that $\ell(t_0) = i(t_0) + 1$ and $\ell(t_1) = i(t_1) + 1$. Then $\ell(t) = \ell(t_0) + \ell(t_1)$, and $i(t) = i(t_0) + i(t_1) + 1$ (we have +1 because of the new root). So

$$
\ell(t) = \ell(t_0) + \ell(t_1)
= (i(t_0) + 1) + (i(t_1) + 1) \quad \text{(IH)}
= (i(t_0) + i(t_1) + 1) + 1
= i(t) + 1
$$
2. (20%) For each of the following statements determine whether it is always true, and explain your answer. You may use major notions and results discussed in class.

(a) Any two infinite sets \( A, B \) are equipollent (i.e. \( A \cong B \)).

**Solution.** False. No set \( A \) is equipollent to its power set \( \mathcal{P}(A) \), So \( \mathbb{N} \) and \( \mathcal{P}(\mathbb{N}) \) are infinite sets that are not equipollent.

(b) Every language denoted by a regular expression without star is finite.

**Solution.** True, by induction on regular expressions. The expressions \( \emptyset, \varepsilon \) and \( \sigma \) (for \( \sigma \in \Sigma \)) denote sets of size \( \leq 1 \); and the operation of union and concatenation yield finite languages from finite languages.

3. (25%) Build a 4-state DFA that recognizes the langauge \( \{ w \in \{a, b, c\}^* | \text{abc is not a substring of } w \} \)

**Suggestion:** Think of your states as “goals”, i.e. “conditions for acceptance.”

**Solution.**
4. (25%) Use the state-elimination algorithm to covert the DFA below to an equivalent regular expression (i.e. the expression denotes the language recognized by the DFA). Show all steps.

Solution.
5. (20%) Show that the language

\[ L = \{ w \in \{a, b, c\}^* \mid \#a(w) + \#b(w) = \#c(w) \} \]

is not regular. You may argue using either residues or clipping. In either case, remember that you can choose strings that are convenient for your argument, and those can be very simple.

**Solution.** (using residues) Consider the residues \( L/a^n = \) for \( n \geq 0 \). For each \( n \geq 0 \) we have \( c^n \in L/a^n+ = \), because for \( w = a^n+ = c^n \) we have \( \#a(w) + \#b(w) = n + 0 = n = \#c(w) \). But \( c^n \notin L/a^i \) for all \( i \neq n \). So all these residues are different. It follows that that \( L \) is not regular.

**Solution.** (using clipping) Given \( k > 0 \), let \( w = a^k+ = c^k \), and let \( u \) be the substring \( c^k \). Then \( w \in L \) and \( |u| \geq k \). But for any clipping of \( w \) in \( u \) we’d get a string \( w' = a^k+ = c^i \) with \( i < k \), which is not in \( L \). So \( L \) fails the clipping property, and cannot be regular.