Problem 3

Let \( x \) and \( y \) be strings and let \( L \) be any language. We say that \( x \) and \( y \) are distinguishable by \( L \) if some string \( z \) exists whereby exactly one of the strings \( xz \) and \( yz \) is a member of \( L \). Otherwise, if for every string \( z \), \( xz \in L \) if and only if \( yz \in L \), we say that \( x \) and \( y \) are indistinguishable by \( L \). If \( x \) and \( y \) are indistinguishable by \( L \) we write \( x \equiv_L y \).

(a) Prove that \( \equiv_L \) is an equivalence relation.

**Reflexivity** want to check: \( x \equiv_L x \).

Let \( xz \in L \). Then \( xz \in L \iff xz \in L \). This is exactly our definition of indistinguishable:

\[ x \equiv_L x. \]

**Symmetry** want to check: if \( x \equiv_L y \) then \( y \equiv_L x \).

Again, by the definition above, if \( x \equiv_L y \) then

\[ xz \in L \iff yz \in L \quad \forall z. \]

Since \( \iff \) is itself symmetric, we can switch the implication above to get

\[ yz \in L \iff xz \in L. \]

This is just the definition of equivalence for \( y \equiv_L x \), which is our desired result.

**Transitivity** we need to check: if \( a \equiv_L b \) and \( b \equiv_L c \), then \( a \equiv_L c \).

We once more return to the definition of distinguishable. If \( a \equiv_L b \), then we can write

\[ az \in L \Rightarrow bz \in L \text{ and } bz \in L \Rightarrow az \in L \quad \forall z, \]

and similarly for \( b \equiv_L c \),

\[ bz \in L \Rightarrow cz \in L \text{ and } cz \in L \Rightarrow bz \in L \quad \forall z. \]

Combining, we get

\[ az \in L \Rightarrow bz \in L \Rightarrow cz \in L \quad \forall z \]

\[ cz \in L \Rightarrow bz \in L \Rightarrow az \in L \quad \forall z, \]

or \( az \in L \Rightarrow cz \in L \) and \( cz \in L \Rightarrow az \in L \). This is the same as saying \( az \in L \iff cz \in L \) which means that \( a \equiv_L c \). (A simpler proof of transitivity also sufficed.)

Because \( \equiv_L \) is reflexive, symmetric and transitive, \( \equiv_L \) is an equivalence relation. \( \Box \)
(b) Take a regular language $L$ and let $X = \{x_1, \ldots, x_k\}$. Suppose $x_i \not\equiv_L y_i$ for all $i, j \leq k, i \neq j$. Prove a DFA that accepts $L$ must have at least $k$ states.  

Suppose $M$ is a DFA which accepts $L$ but has fewer than $k$ states. Then there must be a pair of strings, $x_i, x_j \in X$ such that $M$ is in the same state after reading both $x_i$ and $x_j$. Equivalently, we can write that $\hat{\delta}(q_0, x_i) = \hat{\delta}(q_0, x_j)$.

Take any string $z \in \Sigma^*$. After reading $x_iz$, the DFA must be in the same state as after reading $x_jz$, since it was in the same state before reading $z$. Hence, $M$ must either accept or reject both $x_iz$ and $x_jz$.

Since the DFA $M$ always accepts or rejects $x_iz$ and $x_jz$, we can say

$$x_iz \Leftrightarrow x_jz, \quad \forall z \in \Sigma^*,$$

which means that $x_i \equiv_L x_j$. This contradicts that every pair of strings in $X$ are distinguishable from one another. Hence, $M$ must have at least $k$ states.

A quick note: be careful in treating the start state separately from other states; the start state can often serve more than one purpose.