There are 1 questions for a total of 10 points.

1. (10 points) Let Σ be a finite alphabet. Let $L \subseteq \Sigma^*$ be any language over Σ. Define the relation $=_L$ on Σ^* as follows: $x =_L y$ if and only if for all $z \in \Sigma^*$ either both $x \cdot z$ and $y \cdot z$ are in L, or both are not in L.

   (a) Let $\Sigma = \{0, 1\}$, and let E be the set of all binary strings containing an equal number of 0's and 1's. Prove that the equivalence relation $=_E$ has infinitely many equivalence classes.

   Solution: We shall show this by exhibiting an infinite set of strings such that no two distinct strings in the set are equivalent under $=_E$.
   Notation: by a^n we denote the string which is formed by putting n instances of a symbol $a \in \Sigma$ together. Note that $a^n \in \Sigma^*$.
   Consider the set $S = \{0^n \mid n \in \mathbb{Z}, n > 0\}$.
   Suppose for some $0^n, 0^m \in S$ for $m \neq n$, we have $0^n =_E 0^m$.
   Consider the string $z = 1^n \in \Sigma^*$. Then we note that $0^n \cdot 1^n \in \Sigma^*$ has an equal number of 0s and 1s, and thus belongs to E. However, in the string $0^m \cdot 1^n$, we have m 0s and n 1s, which are not equal, and thus $0^m \cdot 1^n$ is not in E.
   However, this is a contradiction to the definition of $=_E$, and thus our assumption about the existence of such n, m is false, and thus we have shown that for any $m \neq n$, we have $0^m \neq_E 0^n$, implying that there exists one equivalence class for each n (corresponding to 0^n), which implies that the number of equivalence classes is infinite.

   (b) Let $D = (Q, \Sigma, \delta, q_0, A)$ be any DFA and let L be its language. Prove that the number of equivalence classes of $=_L$ is at most $|Q|$.

   Solution: For a state q, define $S_q = \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) = q\}$. Note that the union of all these sets over q is Σ^*, since running a DFA can lead to possibly any of the $|Q|$ states. Also, all of these sets are disjoint.
   Consider the set $W = \{S_q \mid q \in Q \wedge S_q \neq \emptyset\}$. We claim that there is an surjective mapping from W to the set of equivalence classes of $=_L$. This will imply that the number of equivalence classes of $=_L$ is at most $|W| \leq |Q|$, from where we shall be done.
   We will first show that all elements in S_q are equivalent under $=_L$. Note that for $x, y \in S_q$, we have $\hat{\delta}(q_0, x \cdot z) = \hat{\delta}(\hat{\delta}(q_0, x), z) = \hat{\delta}(q, z)$ (the first equality follows from the fact that $x \in S_q$, and the second follows from the definition of $\hat{\delta}$ inductively, the pertaining proof is given in the last paragraph). Similarly, we have $\hat{\delta}(q_0, y \cdot z) = \hat{\delta}(q, z)$. Hence depending on whether $\hat{\delta}(q, z) \in A$ or not, the strings $x \cdot z$ and $y \cdot z$ are either both accepted or rejected by D, which is equivalent to being (respectively) both in or not in L, and thus $x =_L z$.
   We shall now construct a surjective mapping from W to the set of (non-empty, by definition) equivalence classes of $=_L$, which shall solve the problem as claimed above.
   Consider the function f defined as follows:
   $f(S_q) =$ equivalence class of any string in S_q. Note that this mapping is well defined by the third paragraph, and is a total function since each S_q is a function and equivalence classes form a partition of the whole set. Now consider any equivalence class C of S. Since it is non-empty by definition, there exists an element c in C. Now note that $f(S_{\hat{\delta}(q_0,c)}) = C$, which follows from the definition of S_q, and thus for each equivalence class C of $=_L$, we have a corresponding pre-image (under this mapping) of C in W. Thus this function is a surjective function, as needed in the second paragraph.
   * Proof of the fact that $\hat{\delta}(q, x \cdot y) = \hat{\delta}(\hat{\delta}(x), y)$: We proceed by induction on $n = |y|$. Base cases:

$n = 0$: in this case, we have $\hat{\delta}(q, x \cdot y) = \hat{\delta}(q, x \cdot \epsilon) = \hat{\delta}(q, x) = \hat{\delta}(\hat{\delta}(q, x), \epsilon) = \hat{\delta}(\hat{\delta}(q, x), y)$, as needed.

$n = 1$: in this case, we have $\hat{\delta}(q, x \cdot y) = \delta(\hat{\delta}(q, x), y) = \hat{\delta}(\hat{\delta}(q, x), y)$, since y is a single character.

$n > 1$: in this case, let $y = z \cdot a$ for some a of length 1. Then we have $\hat{\delta}(q, x \cdot y) = \hat{\delta}(q, (x \cdot z) \cdot a) = \hat{\delta}(\hat{\delta}(q, x \cdot z), a) = \hat{\delta}(\hat{\delta}(\hat{\delta}(q, x), z), a) = \hat{\delta}(\hat{\delta}(q, x), z \cdot a) = \hat{\delta}(\hat{\delta}(q, x), y)$, as required. The first equality follows by associativity of concatenation, the second by the base case applied on y replaced by a, the third by the base case applied on y replaced by z, the fourth by the base case applied on q replaced by $\hat{\delta}(q, x)$ and x, y replaced by z, a, and the fifth by the equality of y and $z \cdot a$.

(c) Using the above two claims, prove that E is not a regular language.

Solution:

Suppose, on the contrary, that E is in fact a regular language.

Then there must exist a DFA $D = (Q, \Sigma, \delta, q_0, A)$ such that $E = \mathcal{L}(D)$ (note that Q here is a finite set of states by the definition of a DFA).

Then we must have at most $|Q|$ equivalence classes of $=_E$ by Q1.2. However, there are infinitely many equivalence classes of $=_E$ by Q1.1, which is a contradiction, as Q is a finite set of states.

Hence the assumption that E is regular is wrong, and thus E is not a regular language.