1. There can be different automata that accept the same language. In this problem you will walk through an algorithm that converts a given DFA to an equivalent DFA with the minimal number of states. The key idea is that we want to consolidate “duplicate” states; i.e. states that “behave the same”.

Some of these questions refer to the following machine $M_{example}$:

(a) We need to consider the language of a machine starting from a given state. Let $L(M, q)$ be the set of strings that $M$ would accept if it started in state $q$. Write down a formal definition for $L(M, q)$ in terms of the extended transition function.

(b) We can define an equivalence relation $\sim$ to capture the idea that two states “behave the same”: we say that $q_1 \sim q_2$ if $L(M, q_1) = L(M, q_2)$. Prove that $\sim$ is an equivalence relation.

(c) We now want to create a new machine $M'$ such that $L(M') = L(M)$. The set of states of $M'$ will be the quotient of the states of $M$ by the equivalence relation $\sim$ (in other words, $Q' = Q/\sim$). Explicitly write down the set $Q'$ for the machine $M_{example}$ depicted above. (Hint: there are three elements of $Q'$, each of which is a set).

(d) We can define the transition function $\delta'$ for $M'$ as follows: given an equivalence class $q' \in Q'$, we can choose a representative element $q$ of $q'$ and let $\delta'(q', a)$ be the equivalence class of $\delta(q, a)$.

This may be ambiguous, because there may be many different representatives of the equivalence class $q'$. Show that the function is in fact well defined, by showing that if we make two different choices of $q$ we still get the same value $\delta'(q', a)$.

(e) Write down a general expression for the initial state and the set of final states of $M'$.

(f) Draw $M'$ for $M_{example}$. 
