Problem 1  a. Prove that if every state of a DFA $M$ is an accepting state (i.e., machine $M$ has $F = Q$) then $M$ accepts every string (i.e., $L(M) = \Sigma^*$).

b. Is the converse true? In other words, if $L(M) = \Sigma^*$ for some DFA $M$, does it follow that $M$ has $F = Q$?

Problem 2  Consider a DFA $M$ whose only accept state is its start state (i.e., a DFA for which $F = \{q_0\}$). Show that $L(M) = A^*$ for some language $A$.

Note: In fact, in this case $L(M) = A^*$ for some regular language $A$, but showing this is somewhat harder.

Problem 3  We say that a language has epsilon if it includes the empty string. Give an algorithm for deciding whether the language of a DFA has epsilon. That is, explain how, on input a DFA $M = (Q, \Sigma, \delta, q_0, F)$, you can decide whether $M$ accepts $\varepsilon$ or not.

Hint: This problem is easy to solve with DFAs, don’t overthink it!

Problem 4  The partial of a language $L$ (over some alphabet $\Sigma$) with respect to a character $c \in \Sigma$ is written $\partial_c(L)$ and defined as $\partial_c(L) = \{w \mid cw \in L\}$.

Show that if a language $L$ over alphabet $\Sigma$ is regular then so is $\partial_c(L)$ for any $c \in \Sigma$.

Hint: Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA recognizing $L$. Show that some DFA $M' = (Q', \Sigma, \delta', q'_0, F')$ whose components are defined in terms of $M$’s components, recognizes the language $\partial_c(L)$. 
