Algorithms for Regular and CFG Languages

In the following, let $M = (Q, \Sigma, \delta, s, A)$ be a DFA with n states, over an alphabet Σ of constant size.

1 Describe an algorithm for deciding if $L(M) = \emptyset$.

2 Describe an algorithm for deciding if $L(M) = \Sigma^*$.

3 Describe an algorithm for deciding if $L(M)$ is finite.

4 Given two DFAs M, M' decide if $L(M) \subseteq L(M')$.

5 Given two DFAs M, M' decide if $L(M) = \overline{L(M')} = \Sigma^* \setminus L(M')$.

6 Given a CFG $G = (V, T, P, S)$, decide if $L(G)$ contains any string.

7 Two sets $q, q' \in Q$, are distinguishable, if there exists a string $w \in \Sigma^*$, such that $\delta(q, w) \in A$ and $\delta(q', w) \notin A$ (or vice versa). Show how to compute the set D_0 of all the pairs of states that are distinguishable with strings of length 0.

8 Let D_i be all the set of pairs of states of M that are distinguishable with strings of length at most i. Show how to compute D_{i+1} from D_i. (Think about $i = 0$ first, and then $i = 1$, etc.)

9 One can show that if $D_i = D_{i+1}$, then D_i is the set of all distinguishable pairs of states of M. Since $|D_i| \leq \binom{n}{2}$, it follows that this happens after at most $O(n^2)$ iterations of the algorithm using the above steps. Let D^* be the set of pairs the first iteration this happens – this is the set of all distinguishable pairs of states of M. Given M and D^*, show how to compute a minimal automata equivalent to M.