DECISION PROBLEMS FOR REGULAR LANGUAGES

11/12/2009
E8.1

Exercise 1:

Give algorithms to tell whether

a) a given regular language L is universal (i.e. $L = \Sigma^*$).

b) two regular languages have at least one string in common.

Exercise 2:

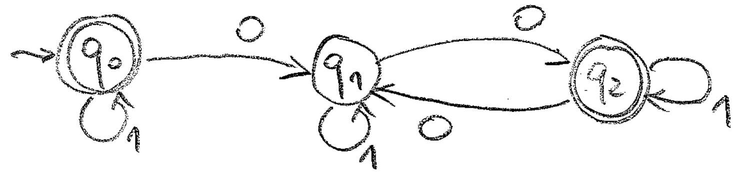
Using the characterization of regular languages in terms of DFAs, show the following:

If L_1 and L_2 are regular, then so is $L_1 \cap L_2$.

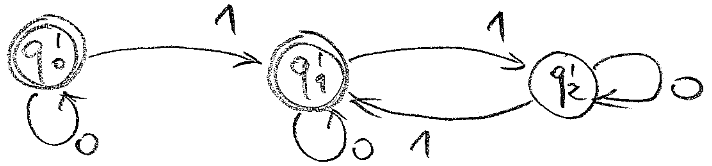
Do not rely on De Morgan's law $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$.

Apply the construction of a DFA for $L_1 \cap L_2$ to the following DFAs A_1 for L_1 and A_2 for L_2:

A_1 :



A_2 :



Note: we can assume that L_1 and L_2 are RLs over the same alphabet Σ.

SOLUTIONS

E8.2

1) a) If $L = \Sigma^*$, then $\bar{L} = \overline{\Sigma^*} = \emptyset$

Hence, we need to check whether $\bar{L}$ is empty.

Algorithm when L is given as a DFA D_L:

1) Construct a DFA $D_{\bar{L}}$ s.t. $\mathcal{L}(D_{\bar{L}}) = \bar{L}$ by swapping final and non-final states of D_L.

2) Check whether $D_{\bar{L}}$ is empty (by constructing the set of states reachable from the initial state, and checking whether it contains at least one final state)

Algorithm when L is given as an NFA N_L:

1) Determinize N_L, i.e. construct a DFA D_L s.t. $\mathcal{L}(D_L) = \mathcal{L}(N_L)$ (Note: D_L might have a number of states that is exponential in the number of states of N_L).

2) Proceed as in the case of a DFA

Algorithm when L is given as a RE E_L:

1) Construct an ε-NFA N_{ε_L} s.t. $\mathcal{L}(N_{\varepsilon_L}) = \mathcal{L}(E_L)$

2) Eliminate ε-transitions from N_{ε_L}, obtaining an NFA N_L s.t. $\mathcal{L}(N_L) = \mathcal{L}(N_{\varepsilon_L})$

3) Proceed as in the case of an NFA

E8.3

1) b) To check whether two RLs L_1 and L_2 have at least one string in common, we can check whether $L_1 \cap L_2$ is nonempty.

Algorithm:

1) Construct a DFA / NFA / ε-NFA / RE for $L_1 \cap L_2$, starting from DFAs/NFAs/ε-NFAs/REs for L_1 and for L_2

2) Check whether $L_1 \cap L_2$ is not-empty.

Note: to construct a DFA / NFA / ε-NFA / RE for $L_1 \cap L_2$, we can use De Morgan's law.

- $L_1 \cap L_2$ is still a RL, since RLs are closed under intersection.

E8.4

2) Let $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ be a DFA s.t. $\mathcal{L}(A_1) = L_1$.
Let $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ — " — $\mathcal{L}(A_2) = L_2$.

Consider a string w accepted by both A_1 and A_2.
Let $w = a_1 a_2 \cdots a_n$. Then we have

in A_1 $\quad q_{01} \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \xrightarrow{a_3} \cdots \xrightarrow{a_n} p_n \in F_1$

in A_2 $\quad q_{02} \xrightarrow{a_1} p_1' \xrightarrow{a_2} p_2' \xrightarrow{a_3} \cdots \xrightarrow{a_n} p_n' \in F_2$

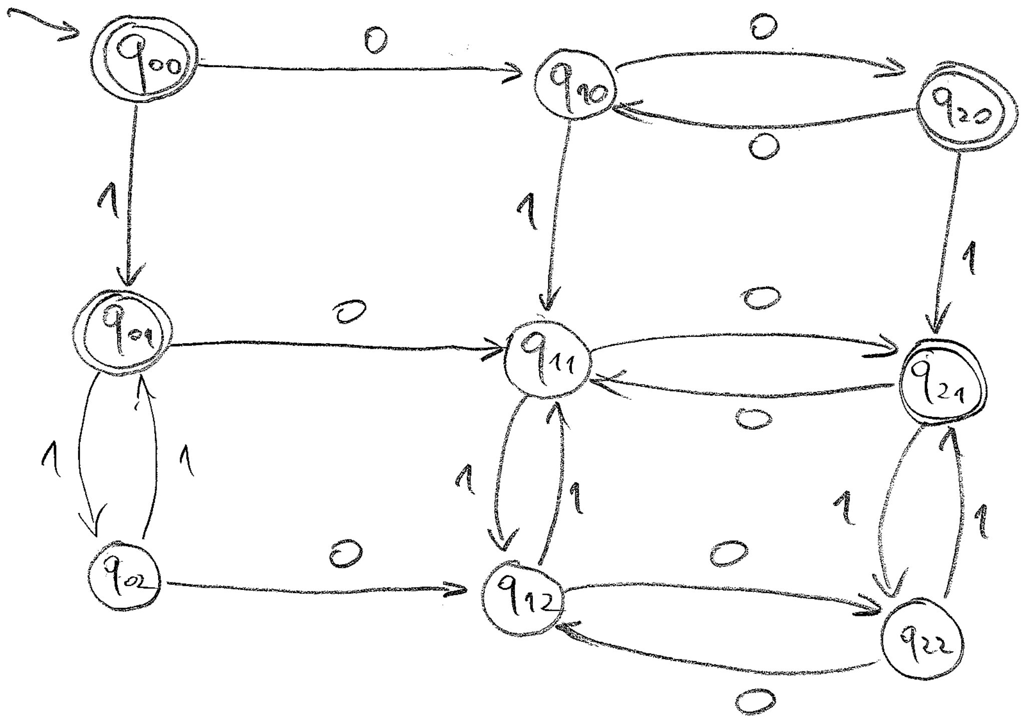
Hence we can construct a DFA $A_\cap = (Q_\cap, \Sigma, \delta_\cap, q_{0\cap}, F_\cap)$ that simulates the transitions of both A_1 and A_2:

- Each state of $A_\cap$ is a pair of states (q_1, q_2), where $q_1 \in Q_1$ and $q_2 \in Q_2$.
  Hence $Q_\cap = Q_1 \times Q_2$.
- The initial state $q_{0\cap}$ is the pair of initial states of Q_1 and Q_2. Hence $q_{0\cap} = (q_{01}, q_{02})$
- The set of final states is such that both A_1 and A_2 accept if $A_\cap$ accepts, Hence $F_\cap = F_1 \times F_2$
- The transition function $\delta_\cap$ simulates the transitions of both A_1 and A_2: If $A_\cap$ is in state (q_1, q_2), then on input a it goes to a state (q_1', q_2'), where $q_1' = \delta_1(q_1, a)$ and $q_2 = \delta_2(q_2, a)$.
  Hence: for all $a \in \Sigma$, $q_1 \in Q_1$, $q_2 \in Q_2$:
  $$\delta_\cap((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a)).$$

One can show that $A_\cap$ constructed in this way accepts $\mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

$A_\cap$ is called the product automaton.

By applying this construction to the automata A_1 and A_2, we obtain



We have used q_{ij} to denote (q_i, q'_j)