Models of Computation (Sheet #1)

Marius Gavrilescu

1. In all of the following q_0 (or q_{00}) is the start state, and # is a garbage state (non-accepting state, all edges from it go to itself).

   $x \to_i y$ means "edge from x to y with label i". $x \to y$ means $x \to_0 y$ and $x \to_1 y$.

   (a) $q_0 \to q_0$ No accepting states.

   (b) $q_0 \to \#$, q_0 accepting state.

   (c) $q_0 \to q_1, q_1 \to q_1$. q_1 accepting state.

   (d) $q_0 \to_0 q_1, q_1 \to q_2, q_2 \to q_1, q_0 \to_1 q_3, q_3 \to q_4, q_4 \to q_3$. Accepting states q_1 and q_4.

   (e) States are q_{ab} with $a \in \{0,1,2\}$ and $b \in \{0,1\}$. We have $q_{ab} \to_0 q_{(a+1)b}$, $q_{ab} \to_1 q_{a(b+1)}$ if the destination state exists. Additionally, we have $q_{2b} \to_0 q_{2b}$, $q_{a1} \to_1 \#$. Accepting states are q_{20} and q_{21}.

   (f) States are q_{ab} with $a \in \{0,1\}$ and $b \in \{0,1,2,3\}$. We have $q_{0b} \to_0 q_{1b}$, $q_{1b} \to_0 q_{0b}$, $q_{a0} \to_1 q_{a1}$, $q_{a1} \to_1 q_{a2}$, $q_{a2} \to_1 q_{a3}$, $q_{a3} \to_1 q_{a3}$. Accepting states are q_{0b} and q_{a2}.

2. (a) Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognise L_1 and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognise L_2.
   Construct $M = (Q, \Sigma, \delta, q_0, F)$ to recognise $L_1 \cap L_2$:
   $Q = Q_1 \times Q_2$
   $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
   $q_0 = (q_1, q_2)$
   $F = F_1 \times F_2$
   This automaton simulates both automatas at the same time, accepting the string only if both automatas would have accepted it.

   (b) Same as above, but change $F = F_1 \times F_2$ to $F = F_1 \times (Q_2 \setminus F_2)$.

3. (a) $q_i \to_1 q_0$, $q_0 \to_0 q_1$, $q_1 \to_0 q_2$, $q_2 \to_0 q_2$. Accepting state q_2.

   (b) No edges. Accepting state q_0.

   (c) $q_0 \to_0 q_1$. Accepting state q_1.

   (d) $q_0 \to_0 q_1, q_0 \to_1 q_2, q_0 \to q_3, q_1 \to q_1, q_1 \to_0 q_3, q_2 \to q_2, q_2 \to_1 q_3$. Accepting state q_3.

4. (a) 1025 states. First 1024 represent bitmasks of digits seen so far, the last (q_{1024}) is the only accepting state.
   In every normal state, if we get a digit whose bit is 0 we set that bit to 1 (= we move to the state with that bit 1 and other bits unchanged), and if we get a digit whose bit is 1 we move to the same state and we move to the accepting state.

   (b) Same states as before, and same transitions except that we move to the accepting state when we get a digit whose bit is 0, not when we get a digit whose bit is 1.

5. (a) States q_A, q_B, q_C. Start in q_A. Edges $q_A \to_A q_A, q_A \to_B q_B, q_A \to_C q_C, q_B \to_B q_B, q_B \to_C q_C, q_C \to_C q_C$. All states are accepting.

   (b) States $q_0 \ldots q_{10}$. Start in q_{10}. Edges $q_i \to_1 q_0$, $q_i \to_0 q_{i+1}$, $q_{10} \to_0 q_{10}$. All states accepting except for q_{10}.

6. (a) Yes, swapping accepting and non-accepting states yields an NFA.

   No, the new NFA does not necessarily recognise the complement of L. Example: the NFA at 3 (ii) whose language is $\{\epsilon\}$, yields a NFA whose language is $\varnothing$ if we swap accepting and non-accepting states.

   (b) Yes. For any regular language there is a DFA that accepts it. If we swap accepting states and non-accepting states on that DFA, the resulting DFA will accept the complement of the language. So the complement of a regular language is also regular.