Technische Universität München **Winter term 2009/10**
I7
Prof. J. Esparza / J. Křetínský / M. Luttenberger **6. November 2009**

Automata and Formal Languages – Homework 3

Due 12.11.2009.

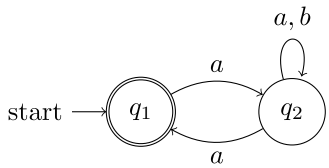
Exercise 3.1

Let $\mathcal{A} = (Q, \Sigma, \delta, q_{\text{initial}}, F)$ be some NFA. We assume that $Q = \{q_1, q_2, \ldots, q_n\}$. With every word $w \in \Sigma^*$ we then associate the boolean matrix $M_w \in \{0,1\}^{n \times n}$ with

$$(M_w)_{i,j} = 1 \text{ iff } \mathcal{A} \text{ can end up in state } q_j \text{ when reading the word } w \text{ starting from state } q_i.$$

The set $\mathcal{T}_\mathcal{A} := \{M_w \mid w \in \Sigma^*\}$ is then called the *transition monoid* of $\mathcal{A}$.

Example: Consider the following FA:



For this automaton, the matrices M_a, M_b, M_{aa} and M_{ab} are:

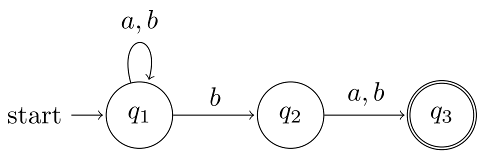
$$M_a = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \; M_b = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \; M_{aa} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \text{ and } M_{ab} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

(a) Assuming standard multiplication of boolean matrices, we have in our example that $M_{aa} = M_a \cdot M_a$ and $M_{ab} = M_a \cdot M_b$.

- Show that $M_u \cdot M_v = M_{uv}$ holds for all $u, v \in \Sigma^*$ for any given NFA $\mathcal{A}$.
- Check that $\mathcal{T}_\mathcal{A}$ w.r.t. this multiplication is indeed a monoid.

  Reminder:
  $\langle S, \cdot \rangle$ is a monoid if (i) $\forall a, b \in S : a \cdot b \in S$, (ii) $\forall a, b, c \in S : a \cdot (b \cdot c) = (a \cdot b) \cdot c$, and (iii) $\exists 1 \in S \forall a \in S : a \cdot 1 = a = 1 \cdot a$.

(b) Consider the following FA $\mathcal{A}$:



- Draw the labeled graph with states the elements of $\mathcal{T}_\mathcal{A}$ and edges $M_u \xrightarrow{a} M_{ua}$ for $a \in \Sigma, u \in \Sigma^*$.

  Note that $\mathcal{T}_\mathcal{A}$ has at most 2^9 elements; construct the graph on the fly starting from M_ε. You should end up with 7 elements/states.
- How can you obtain from this graph a deterministic finite automaton accepting the same language as the original nondeterministic automaton? How does this construction relate to the determinization procedure you have seen in the lecture?

Exercise 3.2

In the preceding exercise, the transition monoid $\mathcal{T}_\mathcal{A}$ of a finite automaton $\mathcal{A}$ has been introduced. Recall also the syntactic monoid $\mathcal{S}_L := \Sigma^* / \equiv_L$ of a language $L \subseteq \Sigma^*$ which was the set of equivalence classes w.r.t. the binary relation $\equiv_L$ on Σ^* defined by

$$x \equiv_L y \text{ iff } \forall u, v \in \Sigma^* : uxv \in L \Leftrightarrow uyv \in L.$$

(a) Let $\mathcal{A} = (Q, \Sigma, \delta, q_i, F)$ be an NFA. Show that for $x, y \in \Sigma^*$ we have

$$M_x = M_y \Rightarrow x \equiv_L y.$$

(b) Let L be a regular language over Σ. Let $\mathcal{A}_L$ be a minimal DFA with $L = \mathcal{L}(\mathcal{A})$. Show that for $x, y \in \Sigma^*$ it holds that

$$x \equiv_L y \Rightarrow M_x = M_y.$$

Is it necessary for $\mathcal{A}$ to be deterministic or minimal?

(c) Calculate the size of the syntatic monoid of the following languages:

- $\mathcal{L}((aa)^*)$ for $\Sigma = \{a\}$.
- $\mathcal{L}((ab + ba)^*)$ for $\Sigma = \{a, b\}$.

You can find some incomplete C++-code on the webpage for an easier calculation of the transition monoid.

(d) A famous result by Schützenberger says that a regular language L is representable by a star-free expression if and only if its syntactic monoid $\mathcal{S}_L$ is aperiodic. (A monoid $\langle M, \cdot, 1\rangle$ is aperiodic if for any $a \in M$ there is a natural number n such that $a^n = a^{n+1}$.)

- Show that syntactic monoid of a regular language L is isomorphic to the transition monoid of a minimal DFA for L, i.e., show:

$$\forall x, y, z \in \Sigma^* \;:\; [x]_L \cdot [y]_L = [z]_L \text{ iff } M_x \cdot M_y = M_z.$$

- Decide for the two languages considered in (c) whether they are representable by star-free expressions. Use a computer.

Exercise 3.3

Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be an NFA and $\sim$ its bisimilarity relation.

(a) Prove or disprove that $\mathcal{L}(q) = \mathcal{L}(q') \Rightarrow q \sim q'$ for $q, q' \in Q$.

(b) A binary relation $R \subseteq Q \times Q$ is called a *simulation* if for any pair $(q, q') \in R$ we have

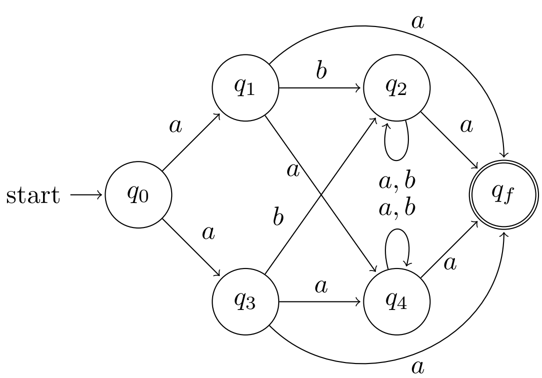
$$(q \in F \Leftrightarrow q' \in F) \wedge \forall a \in \Sigma\, \forall q_a \in \delta(q, a)\, \exists q'_a \in \delta(q', a) \;:\; (q_a, q'_a) \in R.$$

Let $\preceq$ denote $\bigcup\{R \mid R \text{ is a simulation w.r.t. } \mathcal{A}\}$.

- Show that $q \preceq q' \Rightarrow \mathcal{L}(q) \subseteq \mathcal{L}(q')$.
- Prove or disprove: $(q \preceq q' \wedge q' \preceq q) \Rightarrow q \sim q'$.

Exercise 3.4

Consider the following NFA $\mathcal{A}$:



(a) Determine $L := \mathcal{L}(\mathcal{A})$.

(b) Determine the bisimilarity relation $\sim$ of $\mathcal{A}$. Use the partitioning algorithm presented in the lecture.