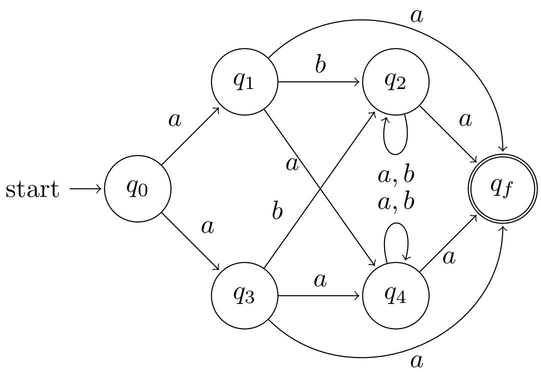
Automata and Formal Languages – Homework 3

Due 12.11.2012.

Exercise 3.1

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



(a) Describe $\mathcal{L}(\mathcal{A})$.

(b) Determine the CSR of $\mathcal{A}$ using the algorithm presented in the lecture.

Exercise 3.2

Consider the partitioning algorithm from the lecture. Its while-loop clearly cannot be executed more than $|Q| - 1$ times. Show that this bound is tight, i.e. give an example where it is executed $|Q| - 1$ times. (Hint: It is sufficient to consider one-letter alphabet.)

Exercise 3.3

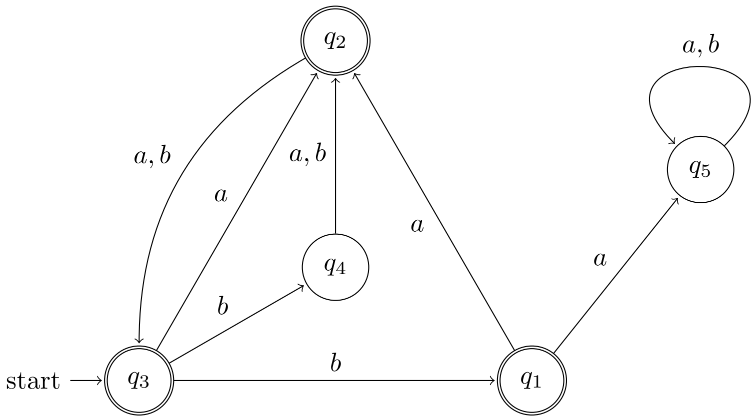
Let $L_i = \{w \in \{a\}^* \mid \text{ the length of } w \text{ is divisible by } i\ \}$.

(a) Construct an NFA for $L := L_4 \cup L_6$ with at most 11 states.

(b) Construct the minimal DFA for L.

Exercise 3.4

Check whether the NFA depicted below recognizes Σ^* by means of the algorithm "UnivNFA" presented in the lecture.



Exercise 3.5

We define the following languages over the alphabet $\Sigma = \{a, b\}$:

- L_1 is the set of all words where between any two occurrences of b's there is at least one a.
- L_2 is the set of all words where every non-empty maximal sequence of consecutive a's has odd length.
- L_3 is the set of all words where a occurs only at even positions.
- L_4 is the set of all words where a occurs only at odd positions.
- L_5 is the set of all words of odd length.
- L_6 is the set of all words with an even number of a's.

Remark: For this exercise we assume that the first letter of a nonempty word is at position 1, e.g., $a \in L_4$, $a \notin L_3$.

Your task is to construct an FA, i.e., DFA or NFA or NFA-ε, for

$$L := (L_1 \setminus L_2) \cup \overline{(L_3 \triangle L_4) \cap L_5 \cap L_6} \text{ where } \triangle \text{ denotes the symmetric difference.}$$

while sticking to the following rules:

- You have to start from FAs for $L_1, \ldots, L_6$.
- Any further automaton has to be constructed from already existing automata via an algorithm introduced in the lecture, e.g. Comp, BinOp, UnionNFA, NFAtoDFA, etc.

Try to find an order on the construction steps which yields an FA for L with as few states as possible.

Exercise 3.6

Since regular languages are closed under complement and intersection, we may extend regular expressions with such primitives: If r and r' are regular expressions, then also $\overline{r}$ and $r \cap r'$ are regular expressions. A language $L \subseteq \Sigma^*$ is called *star-free*, iff there exists an extended regular expression r *without* a Kleene star such that $L = \mathcal{L}(r)$. For example, Σ^* is star-free, because it is the same as $\mathcal{L}(\overline{\emptyset})$. Show that $\mathcal{L}((01 + 10)^*)$ is star-free.