Automata and Formal Languages — Homework 5

Due 21.11.2017

Exercise 5.1
For every $n \in \mathbb{N}$, let $L_n \subseteq \{a, b\}^*$ be the language described by the regular expression $(a+b)^*a(a+b)^n b(a+b)^*$.

(a) Give an NFA A_n with $n+3$ states that accepts L_n.

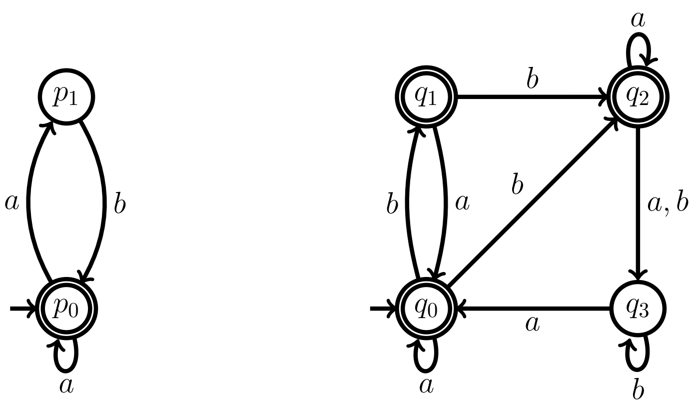
(b) Decide *algorithmically* whether $baabba \in L(A_2)$ and $baabaa \in L(A_2)$.

(c) If you make final and non final states of A_n respectively non final and final, do you obtain an NFA that accepts $\overline{L_n}$? Justify your answer.

(d) Show that $ww \notin L_n$ for every $w \in \{a, b\}^{n+1}$.

(e) Show that any NFA accepting $\overline{L_n}$ has at least 2^{n+1} states. [Hint:]

Exercise 5.2
Consider the following NFAs A and B:



(a) Use algorithm *UnivNFA* to determine whether $L(B) = \{a, b\}^*$.

(b) Use algorithm *InclNFA* to determine whether $L(A) \subseteq L(B)$.

Exercise 5.3

(a) We have seen that testing whether two NFAs accept the same language can be done by using algorithm *InclNFA* twice. Give an alternative algorithm, based on pairings, for testing equality.

(b) Give two NFAs A and B for which exploring only the minimal states of $[\mathit{NFAtoDFA}(A), \mathit{NFAtoDFA}(B)]$ is not sufficient to determine whether $L(A) = L(B)$.

(c) Show that the problem of determining whether an NFA and a DFA accept the same language is PSPACE-hard.