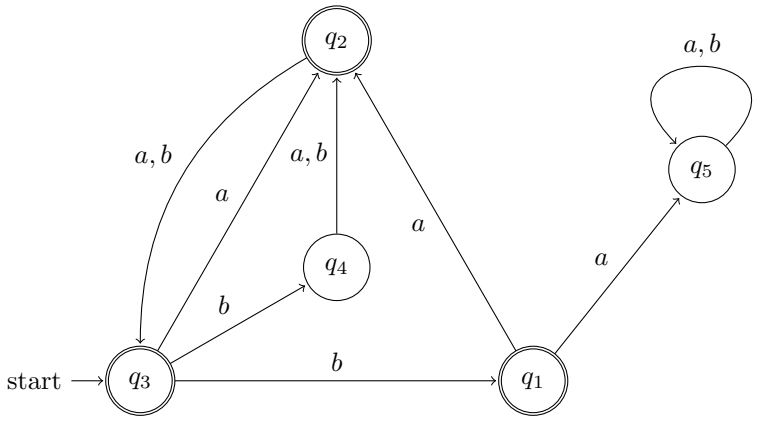
Technische Universität München **Winter term 2015/16**
I7
Prof. J. Esparza / Dr. D. Suwimonteerabuth / Dr. C. Broadbent **06.11.2015**

Automata and Formal Languages — Homework 5

Due 13.11.2015

Exercise 5.1
Let $\Sigma = \{a, b\}$. Check whether the NFA below recognizes Σ^* by means of the algorithm *UnivNFA* presented in the lecture.



Exercise 5.2
Let $\Sigma = \{a, b\}$. Consider the regular expressions $(a + ab)^*$ and $a^*(aba^*)^*$.

a Construct NFAs for the expressions and use *InclNFA* to check if their languages are equal.

b Construct DFAs for the expressions and use *InclDFA* to check if their languages are equal.

c Construct minimal DFAs for the expressions and check whether they are isomorphic.

Exercise 5.3
In the lecture, you have seen that we can save on space using the *lazy* DFAs. However, this does not come for free. There is a space vs. running-time trade-off because of extra steps with head not moving in the case with lazy DFAs.

Find a word w and a pattern p such that the run of the DFA B_p on w takes at most n steps and the run of the lazy DFA C_p takes at least $2n - 1$ steps.

Hint: a simple pattern of the form a^k is sufficient.

Exercise 5.4
Given $w, w' \in \Sigma^*$, we say that w is a *cyclic rotation* of w' if there are some words $w'_1, w'_2 \in \Sigma^*$ such that $w' = w'_1 w'_2$ and $w = w'_2 w'_1$. For instance, *trio* and *riot* are cyclic rotations of each other.

Design a linear-time algorithm to determine whether a word is a cyclic rotation of another.