Technische Universität München **Winter term 2009/10**
I7
Prof. J. Esparza / J. Křetínský / M. Luttenberger **30. November 2009**

Automata and Formal Languages – Homework 8

Due 7.1.2010.

Exercise 8.1

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 $\textbf{wordDFA}(p)$ on w takes at most n steps and the run of $\textbf{lazyDFA}(p)$ takes at least $2n-1$ steps.

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

Exercise 8.2

Design an algorithm that solves the following problem for a finite alphabet Σ. Discuss the complexity of your solution.

- *Given*: $w \in \Sigma^*$ and a regular expression r over Σ.
- *Find*: A shortest prefix $w_1 \in \Sigma^*$ of w such that there exists a prefix $w_1 w_2$ of w and $w_2 \in \mathcal{L}(r)$.