Technische Universität München **Winter term 2011/12**
I7
Prof. J. Esparza / J. Křetínský **13.10.2011**

Automata and Formal Languages – Homework 1

Due 20.10.2011.

Exercise 1.1

We say that $u = a_1 \cdots a_n$ is a *scattered subword* of w (short: $u \lhd w$) if $w = w_0 a_1 w_1 a_2 \cdots a_n w_n$ for some $w_0, \cdots, w_n \in \Sigma^*$.

- Let $L \subseteq \Sigma^*$ be a regular language. Show that $L_\rhd := \{u \in \Sigma^* \mid \exists w \in L : w \lhd u\}$ is also regular.
- Let $L \subseteq \Sigma^*$ be a regular language. Show that $L_\lhd := \{u \in \Sigma^* \mid \exists w \in L : u \lhd w\}$ is also regular.

Exercise 1.2

Let Σ_1, Σ_2 be two alphabets. A map $h : \Sigma_1^* \to \Sigma_2^*$ is called a *homomorphism* if it respects the empty word and concatenation, i.e.,

$$h(\varepsilon) = \varepsilon \text{ and } h(w_1 w_2) = h(w_1)h(w_2) \text{ for all } w_1, w_2 \in \Sigma_1^*.$$

Assume that $h : \Sigma_1^* \to \Sigma_2^*$ is a homorphism. Note that h is completely determined by its values on Σ_1.

(a) Let $\mathcal{A}$ be a finite automaton over the alphabet Σ_1. Describe how to constuct a finite automaton accepting the language

$$h(\mathcal{L}(\mathcal{A})) := \{h(w) \mid w \in \mathcal{L}(\mathcal{A})\}.$$

(b) Let $\mathcal{A}'$ be a finite automaton over the alphabet Σ_2. Describe how to construct a finite automaton accepting the language

$$h^{-1}(\mathcal{L}(\mathcal{A}')) := \{w \in \Sigma_1^* \mid h(w) \in \mathcal{L}(\mathcal{A}')\}.$$

(c) Recall that the language $\{0^n 1^n \mid n \in \mathbb{N}\}$ is context free, but not regular. Use the preceding results to show that $\{(01^k 2)^n 3^n \mid k, n \in \mathbb{N}\}$ is also not regular.

Exercise 1.3

Go to `http://www.cs.duke.edu/csed/jflap/` and download JFLAP. Run it and select the finite automata mode.

- Create a non-deterministic automaton and determinize it.

Determinization (NFAtoDFA) can cause an exponential blowup. We now examine the special case in which the alphabet has only one letter.

- Show that for any n, there is a NFA with $2n$ states so that any DFA recognizing the same language has at least $n(n-1)$ states.

  Hint: Since a word over a singleton alphabet is given by its length, consider e.g. FAs accepting words of lengths divisible by some constants.
- Using JFLAP, verify the correctnes of your example for e.g. $n = 5$.
- Can you also find an example where the DFA has $\mathcal{O}(n^7)$ states?

Exercise 1.4

During the ε-removal (NFAεtoNFA), no transition is ever again added to the worklist *after* it has been added to the worklist, processed *and* removed from the worklist.

- Give an example of an NFA-ε and a run of the ε-removal algorithm where a transition is put into the worklist twice.