Technische Universität München **Winter term 2020/21**
I7
Prof. J. Esparza / M. Lazić / C. Weil-Kennedy

Automata and Formal Languages — Exercise Sheet 5

Exercise 5.1
Let $L_1 = \{baa, aaa, bab\}$ and $L_2 = \{baa, aab\}$.

(a) Give an algorithm for the following operation:

INPUT: A fixed-length language $L \subseteq \Sigma^k$ described explicitly as a set of words.
OUTPUT: State q of the master automaton over Σ such that $L(q) = L$.

(b) Use the previous algorithm to build the states of the master automaton for L_1 and L_2.

(c) Compute the state of the master automaton representing $L_1 \cup L_2$.

(d) Identify the kernels $\langle L_1 \rangle$, $\langle L_2 \rangle$, and $\langle L_1 \cup L_2 \rangle$.

Exercise 5.2
(a) Give an recursive algorithm for the following operation:

INPUT: States p and q of the master automaton.
OUTPUT: State r of the master automaton such that $L(r) = L(p) \cdot L(q)$.

Observe that the languages $L(p)$ and $L(q)$ can have different lengths. Try to reduce the problem for p, q to the problem for p^a, q.

(b) Give an recursive algorithm for the following operation:

INPUT: A state q of the master automaton.
OUTPUT: State r of the master automaton such that $L(r) = L(q)^R$

where R is the reverse operator.

(c) A *coding* over an alphabet Σ is a function $h \colon \Sigma \mapsto \Sigma$. A coding h can naturally be extended to a morphism over words, i.e. $h(\varepsilon) = \varepsilon$ and $h(w) = h(w_1)h(w_2) \cdots h(w_n)$ for every $w \in \Sigma^n$. Give an algorithm for the following operation:

INPUT: A state q of the master automaton and a coding h.
OUTPUT: State r of the master automaton such that $L(r) = \{h(w) : w \in L(q)\}$.

Can you make your algorithm more efficient when h is a permutation?

Exercise 5.3
Let $k \in \mathbb{N}_{>0}$. Let flip : $\{0,1\}^k \to \{0,1\}^k$ be the function that inverts the bits of its input, e.g. flip(010) = 101. Let val : $\{0,1\}^k \to \mathbb{N}$ be such that val(w) is the number represented by w in the *least significant bit first* encoding.

(a) Describe the minimal transducer that accepts

$$L_k = \left\{ [x, y] \in (\{0,1\} \times \{0,1\})^k \mid \text{val}(y) = \text{val}(\text{flip}(x)) + 1 \bmod 2^k \right\}.$$

(b) Build the state r of the master transducer for L_3, and the state q of the master automaton for $\{010, 110\}$.

(c) Adapt the algorithm *pre* seen in class to compute *post* and compute using this algorithm *post*(r, q).