Fachbereich Mathematik
Kord Eickmeyer

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Winter Term 2015/16

Exercise Sheet no. 1 – Formal Foundations of Computer Science

(E1.1) [warm-up: bracketing and regular expressions]

Give all $\alpha \in REG(\{0,1\})$ which, after removing all parentheses, lead to 0+10∗. Which is the one we mean by 0+10∗? What are the languages $L(\alpha)$ for these α?

(E1.2) [Languages] Let Σ be some finite alphabet. Prove or disprove the following identities for arbitrary languages $L_1, L_2 \subseteq \Sigma^*$:

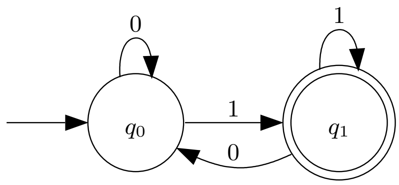
(i) $(L_1 \cup L_2)^* = (L_1^* L_2^*)^*$

(ii) $(L_1 L_2)^* \setminus \{\varepsilon\} = L_1(L_2 L_1)^* L_2$

(iii) $(L_1 L_2)^*(L_1 L_2) = L_1(L_2 L_1)^* L_2$

(E1.3) [a sample DFA]

Consider the following DFA over the alphabet $\Sigma = \{0, 1\}$:



(i) Write this automaton formally as a tuple $(Q, \Sigma, \delta, q_0, F)$, i.e. give the sets Q and F, the initial state q_0 and the transition function δ.

(ii) Describe in words the language recognised by this automaton.

(iii) Give a regular expression describing the same language.

(E1.4) [regular languages]

For each of these languages, give a regular expression E_i and a deterministic finite automaton A_i such that $L_i = L(E_i) = L(A_i)$:

$L_1 := \{w \in \{a, b, c\}^* \mid |w|_a \geq 1 \text{ and } |w|_b \geq 2\}$,
$L_2 := \{w \in \{0, 1\}^* \mid \text{there is at most one pair of adjacent 1s in } w\}$,
$L_3 := \{w \in \{0, 1\}^* \mid \text{every pair of adjacent 0s appears before any pair of adjacent 1s}\}$.

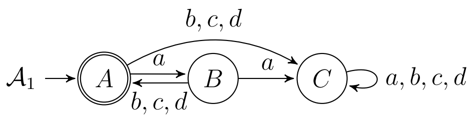
(E1.5) [converting NFAs to DFAs]

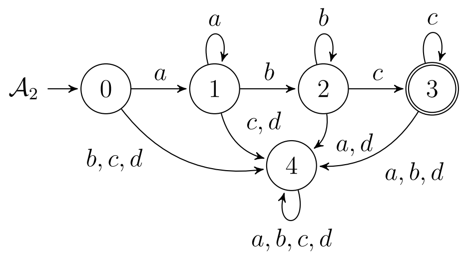
Sketch a transition diagram for the NFA $A := (\{p,q,r,s,t\}, \{0,1\}, \tilde{\Delta}, p, \{s,t\})$, with $\tilde{\Delta}$ as below. Convert it to a DFA, sketch the resulting transition diagram. Informally describe the language it accepts.

$\delta \quad :=$

	p	q	r	s	t
0	$\{p,q\}$	$\{r,s\}$	$\{p,r\}$	$\emptyset$	$\emptyset$
1	$\{p\}$	$\{t\}$	$\{t\}$	$\emptyset$	$\emptyset$

(E1.6) [Product automata] Consider the following two deterministic finite automata:





Using the product automata construction, give a DFA that accepts $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$.