ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Prof. Dr. Andreas Podelski
Matthias Heizmann
Christian Schilling

May 20th, 2014

1. Presence Exercise Sheet for the Lecture
Computer Science Theory

Exercise 1: Sets

(a) Two sets are equal if and only if ____.

Write down all elements (without the duplicates) for the twelve sets below.

(1): __________	(2): __________	(3): __________
(4): __________	(5): __________	(6): __________
(7): __________	(8): __________	(9): __________
(10): __________	(11): __________	(12): __________

Draw lines between those sets which are equal.

(1) $\emptyset$

(2) $\{\diamondsuit, \heartsuit\}$

(3) $\{\}$

(4) $\{\diamondsuit\heartsuit\}$

(5) $\{\{\clubsuit\}\}$

(6) $\{\{\}, \emptyset\}$

(7) $\{\clubsuit, \clubsuit\}$

(8) $\{\{\}\}$

(9) $\{\heartsuit, \diamondsuit\}$

(10) $\{\emptyset\}$

(11) $\{\clubsuit\}$

(12) $\{\emptyset, \diamondsuit\}$

(b) Apply the following set operations and give the number (i), yes/no (ii–iii), and the resulting sets (iv–vi).

(i) $|S|$ for finite set S is defined as ________________________________.

$$|\{\}| = \_\_\_\_\_$$
$$|\{\heartsuit, \clubsuit\}| = \_\_\_\_\_$$
$$|\{\{\}, \{\diamondsuit\}\}| = \_\_\_\_\_$$
$$|\{\{\heartsuit, \clubsuit\}\}| = \_\_\_\_\_$$

(ii) $e \in S$ if and only if ____.

$\{\} \in \{\}$ _____

$\{\} \in \{\heartsuit, \{\}\}$ _____

$\diamondsuit \in \{\heartsuit, \{\diamondsuit\}\}$ _____

$\{\heartsuit\} \notin \{\heartsuit, \{\diamondsuit\}\}$ _____

(iii) $S_1 \subseteq S_2$ if and only if ____________________________________.

$\{\} \subseteq \{\}$ _____

$\{\} \subseteq \{\{\diamondsuit\}\}$ _____

$\{\diamondsuit, \clubsuit\} \subseteq \{\diamondsuit, \heartsuit, \{\clubsuit\}\}$ _____

$\{\diamondsuit, \clubsuit\} \subseteq \{\clubsuit, \heartsuit, \diamondsuit\}$ _____

(iv) $S_1 \cup S_2$ is the set which ____________________________________.

$$\{\} \cup \{\} = \_\_\_\_\_\_\_\_\_\_$$
$$\{\} \cup \{\heartsuit\} = \_\_\_\_\_\_\_\_\_\_$$
$$\{\diamondsuit, \heartsuit\} \cup \{\heartsuit, \clubsuit\} = \_\_\_\_\_\_\_\_\_\_$$

(v) $S_1 \cap S_2$ is the set which ____________________________________.

$$\{\} \cap \{\} = \_\_\_\_\_\_\_\_\_\_$$
$$\{\} \cap \{\heartsuit\} = \_\_\_\_\_\_\_\_\_\_$$
$$\{\diamondsuit, \heartsuit\} \cap \{\heartsuit, \clubsuit\} = \_\_\_\_\_\_\_\_\_\_$$

(vi) $S_1 \setminus S_2$ is the set which ____________________________________.

$$\{\clubsuit\} \setminus \{\} = \_\_\_\_\_\_\_\_\_\_$$
$$\{\} \setminus \{\heartsuit\} = \_\_\_\_\_\_\_\_\_\_$$
$$\{\diamondsuit, \heartsuit\} \setminus \{\heartsuit, \clubsuit\} = \_\_\_\_\_\_\_\_\_\_$$

Exercise 2: Natural numbers as a language
Consider the alphabet $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Give a definition for a language L over Σ containing exactly all natural numbers ($\mathbb{N}$) without leading zeros. This means we do not want to have 1 and 001 (but only 1).

Hint: You can define the language directly or you can apply set operations.
Ask yourself: are we interested in how many leading zeros a word has?
Do not forget 0 (zero).

Exercise 3: Deterministic finite automata
Consider the following picture:

We start at ① and want to get to ③. We can move from ① to ②, from ② to both ① and ③, and from ③ to ②.

(a) Your task is to represent the language of all valid moves from ① to ③ as a DFA.

(i) Let $\Sigma = \{r\}$. For each r we go one step to the right.
(ii) Let $\Sigma = \{\ell\}$. For each ℓ we go one step to the left.
(iii) Let $\Sigma = \{r, \ell\}$.

(b) How many words are accepted in each case?

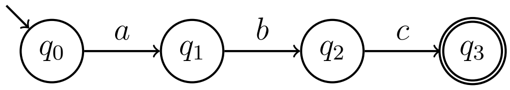
(c) How can we modify the automaton from (iii) if

(i) we start at ②?
(ii) we want to get to ② instead?
(iii) we want to get to ② or ③?

Exercise sheet 3

Exercise 1: Reverse Operator
Consider $\Sigma = \{a, b, c\}$.

(a) What is the language $L(\mathcal{A})$ and its reverse language $L(\mathcal{A})^R$ for the NFA $\mathcal{A}$ below?



$L(\mathcal{A}) =$ ______________ $L(\mathcal{A})^R =$ ______________

Construct an NFA that recognizes the reverse language $L(\mathcal{A})^R$.

(b) What is the problem with the construction if we have more than one final state?

Exercise 2: Regular Expressions
Construct regular expressions for the following languages over the alphabet $\Sigma = \{a, b\}$.

(a) $L_1 = \{a, b, ab\}$

(b) $L_2 = \Sigma^*$

(c) $L_3 = \Sigma^+$

(d) $L_4 = \{w \in \Sigma^* \mid w \text{ starts with } a\}$

Exercise 3: Pumping Lemma
The proof always works as follows:

(a) Assume the language L is regular. Then the pumping lemma must hold.

(b) Assume some $n \in \mathbb{N}$ from the pumping lemma. You must *not* make any assumptions on n.

(c) Smartly choose a word $z \in L$ (usually depending on n) with $|z| \geq n$.

(d) Assume some decomposition $z = uvw$ (with the rules given in the pumping lemma).

(e) Smartly choose some $i \in \mathbb{N}$ such that $uv^iw \notin L$ (often $i = 0$ or $i = 2$ suffices).

With this "algorithm" in mind, read the example provided in the lecture script on page 27.

In general, you may need to make a case distinction in step (d). Then for each case you have to find some i in the next step. But in this exercise it is not necessary.