Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev

Theoretical Computer Science - WS13/14
Exercise Sheet 1 (due: 24.02.14, 40 points)

Organizational notes:

- Please, register for the lecture on our website and in the HISPOS.
- Exercise sheets will be handed out every Mon/Wed before the lecture and the solutions will be collected next Mon/Wed also before the lecture.
- You are allowed to solve the exercises in groups of up to 3 students. Groups should not change over the semester.
- You need to get 50% of all exercise points in order to be admitted to the exam.

Exercise 1: (regular expressions) (2 + 3 + 6 points)
Two regular expressions are called *equivalent* (or isomorphic) if they produce the same language. Proof that the following equivalences hold for regular expressions c, d, e, and f:

1. $(c \cup d) \cup e = c \cup (d \cup e)$
2. $(c \cup d) \cdot e = c \cdot e \cup d \cdot e$
3. $(e \cup f)^* = (e^* \cdot f^*)^*$

Hint: split the proof for the last equivalence into two parts: $L((e \cup f)^*) \subseteq L((e^* \cdot f^*)^*)$ and $L((e^* \cdot f^*)^*) \subseteq L((e \cup f)^*)$

Exercise 2: (deterministic finite automata (DFA) and regular expressions) (4 + 3 + 3 points)

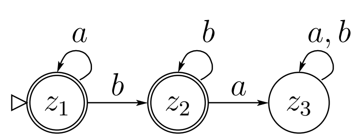
1. Construct a DFA which accepts the following language:

$$L = \{x \in \{a, b\}^* \mid \text{the first symbol in } x \text{ is } a \text{ or the last symbol in } x \text{ is } b\}.$$

   Draw a state diagram and give a short explanation.

2. Provide regular expressions which define the following languages:
   - $L_a = \{w \in \{a, b\}^* \mid w \text{ contains an even number of } a\text{'s}\}$
   - $L_b = \{w \in \{a, b\}^* \mid |w| \text{ is divisible by } 3\}$

Exercise 3: (DFA and regular expressions) (3 points)
Consider the following DFA M:



Give a regular expression e, such that $L(e) = L(M)$.

Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev

Theoretical Computer Science - WS13/14
Exercise Sheet 1 (due: 24.02.14, 40 points)

Exercise 4: (NDFA and regular languages) (3 + 3 points)
In the lecture we used induction on the structure of regular expressions to prove that every regular language is accepted by a nondeterministic finite automaton (NDFA):

$$L = L(e) \wedge e \in reg(A) \Longrightarrow \exists M : L = L(M).$$

Prove the cases which were left out in the lecture (i.e., construct NDFA M):

- if L_1 is accepted by NDFA M_1 and L_2 is accepted by NDFA M_2, then there exists NDFA M, s.t. $L_1 \cup L_2$ is accepted by M,
- if L_1 is accepted by NDFA M^1, then there exists NDFA M, s.t. $(L_1)^*$ is accepted by M.

Exercise 5: (the 'pigeonhole' principle) (5 points)
The 'pigeonhole' principle says that if $n+1$ pigeons are sitting on n holes, then at least one hole should contain more than one pigeon. The same argument can be applied to a DFA $M = (Z, A, \delta, z_0, Z_a)$ with n states: in every computation with length $n+1$ at least one state is encountered more than once:

$$\forall n \in \mathbb{N}_0, (z_i \in Z), w, w' \in A^*.\ (z_0, w) \vdash \cdots \vdash (z_n, w') \wedge \#Z = n \Longrightarrow \exists i, j \leq n.\ i \neq j \wedge z_i = z_j$$

Prove this property by induction on $n \in \mathbb{N}$.

Exercise 6: (NDFA) (5 points)
Consider NDFA $M = (A, Z, \delta, z_0, Z_a)$ and DFA $M' = (A, Z', \delta', z_0', Z_a')$, which is constructed out of M in such a way that

$$Z' = 2^Z \qquad \delta'(z', a) = \bigcup_{x \in z'} \delta(x, a) \qquad z_0' = \{z_0\} \qquad z' \in Z_a' \Leftrightarrow z' \cap Z_a \neq \emptyset$$

It follows from the construction that

$$(z_0, w) \vdash_M^n (z, w') \Longrightarrow \exists z' : (z_0', w) \vdash_{M'}^n (z', w').$$

Prove that $L(M) = L(M')$. You can use the property stated in the lecture:

$$(z_0', w) \vdash_{M'}^n (z', w') \implies z' = \{z \in Z \mid (z_0, w) \vdash_M^n (z, w')\}.$$