**Technische Universität München** **Winter term 2015/16**
**I7**
**Prof. J. Esparza / Dr. D. Suwimonteerabuth / Dr. C. Broadbent** **20.11.2015**

# Automata and Formal Languages — Homework 7

Due 27.11.2015

### Exercise 7.1

We have defined transducers as finite automata whose transitions are labeled by pairs of symbols $(a, b) \in \Sigma \times \Sigma$. With this definition transducers can only accept pairs of words $(a_1 \dots a_n, b_1 \dots b_n)$ of the same length. In many applications this is limiting.

An *$\varepsilon$-transducer* is an NFA whose transitions are labeled by elements of $(\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\})$. An $\varepsilon$-transducer accepts a pair $(w, w')$ of words if it has a run

$$q_0 \xrightarrow{(a_1,b_1)} q_1 \xrightarrow{(a_2,b_2)} \cdots \xrightarrow{(a_n,b_n)} q_n \text{ with } a_i, b_i \in \Sigma \cup \{\varepsilon\}$$

such that $w = a_1 \dots a_n$ and $w' = b_1 \dots b_n$. Note that $|w| \leq n$ and $|w'| \leq n$. The relation accepted by the $\varepsilon$-transducer $T$ is denoted by $L(T)$.

(a) Construct $\varepsilon$-transducers $T_1, T_2$ such that $L(T_1) = \{(a^n b^m, c^{2n}) \mid n, m \geq 0\}$, and $L(T_2) = \{(a^n b^m, c^{2m}) \mid n, m \geq 0\}$.

(b) Show that no $\varepsilon$-transducer recognizes $L(T_1) \cap L(T_2)$.

### Exercise 7.2

Transducers can be used to capture the behaviour of simple programs. The figure below shows a program $P$, where the instruction **end** finishes the execution of the program and ? denotes a non-deterministic value. The domain of the variables is assumed to be $\{0, 1\}$, and the initial value is assumed to be 0.

```
1 x ← ?
2 write x
3 while true do
      repeat
          read y
4     until y = x ∧ y
5     if x = y then
6         write y end
7     if ? then x ← x − 1
      else y ← x + y
8     if x ≠ y then
9         write x end
```

Let $[i, x, y]$ denote the configuration of $P$ in which $P$ is at line $i$ of the program, and the values of $x$ and $y$ are x and y, respectively. The initial configuration of $P$ is $[1, 0, 0]$. By executing the first instruction $P$ moves nondeterministically to one of the configurations $[2, 0, 0]$ and $[2, 1, 0]$; no input symbol is read and no output symbol is written. Hence, the transition relation $\delta$ for $P$ contains the transition rules $([1, 0, 0], (\varepsilon, \varepsilon), [2, 0, 0])$ and $([1, 0, 0], (\varepsilon, \varepsilon), [2, 1, 0])$. Similarly, by executing its second instruction, the program $P$ moves from $[2, 1, 0]$ to $[3, 1, 0]$ while reading nothing and writing 1. Hence, $\delta$ contains $([2, 1, 0], (\varepsilon, 1), [3, 1, 0])$

(a) Draw an $\varepsilon$-transducer that characterizes the program $P$

(b) Can an overflow error occur?

(c) What are the possible values of x upon termination, i.e. upon reaching **end**?

(d) Is there an execution during which $P$ reads 101 and writes 01?

(e) Let $I$ and $O$ be regular sets of inputs and outputs, respectively. Think of $O$ as a set of dangerous outputs that we want to avoid. We wish to prove that the inputs from $I$ are safe, i.e. that when $P$ is fed inputs from $I$, none of the dangerous outputs can occur. Describe an algorithm that decides, given $I$ and $O$, whether there are $i \in I$ and $o \in O$ such that $(i, o)$ belongs to the $I/O$-relation of $P$.

**Exercise 7.3**

Let $\Sigma = \{0, 1\}$. Given $a, b \in \Sigma$, let $a \cdot b$ be the usual multiplication (an analog of boolean *and*) and let $a \oplus b$ be 0 if $a = b = 0$ and 1 otherwise (an analog of boolean *or*)

Consider the boolean function $f : \Sigma^6 \to \Sigma$ defined by

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = (x_1 \cdot x_2) \oplus (x_3 \cdot x_4) \oplus (x_5 \cdot x_6)$$

(a) Construct the minimal DFA recognizing $\{x_1x_2x_3x_4x_5x_6 \mid f(x_1, x_2, x_3, x_4, x_5, x_6) = 1\}$.
(For instance, the DFA accepts 111000 because $f(1, 1, 1, 0, 0, 0) = 1$, but not 101010, because $f(1, 0, 1, 0, 1, 0) = 0$.)

(b) Construct the minimal DFA recognizing $\{x_1x_3x_5x_2x_4x_6 \mid f(x_1, x_2, x_3, x_4, x_5, x_6) = 1\}$.
(Notice the different order!)

(c) More generally, consider the function

$$f(x_1, \ldots, x_{2n}) = \bigoplus_{1 \leq k \leq n} (x_{2k-1} \cdot x_{2k})$$

and the languages $\{x_1x_2 \ldots x_{2n-1}x_{2n} \mid f(x_1, \ldots, x_{2n}) = 1\}$ and $\{x_1x_3 \ldots x_{2n-1}x_2x_4 \ldots x_{2n} \mid f(x_1, \ldots, x_{2n}) = 1\}$.
Show that the size of the minimal DFA grows linearly in $n$ for the first language, and exponentially in $n$ for the second language.