Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
Mohamad Ahmadi

Theoretical Computer Science - Bridging Course
Winter Term 2019/2020
Exercise Sheet 7

for getting feedback submit electronically by 12:15 am, Monday, December 9, 2019

Exercise 1: Decidability *(7 Points)*

Let Σ be a fixed finite alphabet. Show that the language of deterministic finite automatons (DFAs) on Σ that accept every word is decidable. Formally, show that

$$L = \{\langle A\rangle \mid A \text{ is a deterministic finite automaton with } L(A) = \Sigma^*\}$$

is a decidable language.

Remark: You can use that it is not difficult to construct a Turing machine which tests whether an input is the well formed encoding of a deterministic finite automaton.

Sample Solution

Let B be a DFA such that the language generated by B is Σ^*. That is, $L_B = \Sigma^*$. (It is easy to see that this is always possible for any given alphabet Σ.) We have shown in the video lecture that testing equivalence for two DFA is a decidable problem. Let M be such a turing machine that can test equivalence for two DFA. We can construct a turing machine M', such that upon input A where A is a DFA, it will run M with input $\langle A, B\rangle$. If M accepts the input, then M' accepts the input A as well, otherwise M' rejects. Since M will give definite answer in finite time, we know M' will give definite answer in finite time as well. Hence, we know $\mathcal{L}$ is decidable.

Exersive 2: Landau Notation *(2+2+3 Points)*

The set $\mathcal{O}(f)$ contains all functions that are asymptotically not growing faster than the function f (when additive or multiplicative constants are neglected). That is:

$$g \in \mathcal{O}(f) \iff \exists c \geq 0, \exists M \in \mathbb{N}, \forall n \geq M : g(n) \leq c \cdot f(n)$$

For the following pairs of functions, check whether $f \in \mathcal{O}(g)$ or $g \in \mathcal{O}(f)$ or both. Prove your claims (you do not have to prove a negative result $\notin$, though).

(a) $f(n) = 100n,\ g(n) = 0.1 \cdot n^2$

(b) $f(n) = \sqrt[3]{n^2},\ g(n) = \sqrt{n}$

(c) $f(n) = \log_2(2^n \cdot n^3),\ g(n) = 3n$ ***Hint:*** *You may use that* $\log_2 n \leq n$ *for all* $n \in \mathbb{N}$.

Sample Solution

(a) It is $100n \in \mathcal{O}(0.1n^2)$. To show that we require constants c, M such that $100n \leq c \cdot 0.1n^2$ for all $n \geq M$. Obviously this is the case for $c = 1000$ and $M = 1$.

(b) We have $g(n) \in O(f(n))$. Let $c := 1$ and $M := 1$. Then we have

$$
\begin{aligned}
& g(n) \leq c \cdot f(n) && (1)\\
\Leftrightarrow \quad & \sqrt{n} \leq n^{2/3} && (2)\\
\Leftrightarrow \quad & 1 \leq n^{1/6} && (3)\\
\Leftrightarrow \quad & 1 \leq n && (4)
\end{aligned}
$$

The last inequality is satisfied because $n \geq M = 1$.

(c) $f(n) \in O(g(n))$ holds. We give $c > 0$ and $M \in \mathbb{N}$ such that for all $n \geq M$: $\log_2(2^n \cdot n^3) \leq c \cdot n$. As $cn \in O(g(n))$ holds for every constant $c > 0$ the result will follow with the transitivity of the O-notation.

$$
\begin{aligned}
& \log_2(2^n \cdot n^3)\\
&= \log_2(2^n) + \log_2(n^3)\\
&= n + 3 \cdot \log_2(n)\\
&\leq n + 3n = 4n.
\end{aligned}
$$

Thus $\log_2(2^n \cdot n^3) \leq c \cdot n$ for $n \geq M := 1$ and $c := 4$.

We also have that $g(n) \in O(f(n))$ holds because

$$g(n) = 3n \leq 3(n + 3 \cdot \log_2(n)) = 3(\log_2(2^n \cdot n^3)) = 3 \cdot f(n).$$

Thus with $c = 3$ and for $n \geq M := 1$ we have $g(n) \leq cf(n)$.

Exercise 3: Sorting Functions by Asymptotic Growth *(6 Points)*

Sort the following functions by asymptotic growth using the $\mathcal{O}$-notation. Write $g <_{\mathcal{O}} f$ if $g \in \mathcal{O}(f)$ *and* $f \notin \mathcal{O}(g)$. Write $g =_{\mathcal{O}} f$ if $f \in \mathcal{O}(g)$ *and* $g \in \mathcal{O}(f)$.

n^2	$\sqrt{n}$	2^n	$\log(n^2)$
3^n	n^{100}	$\log(\sqrt{n})$	$(\log n)^2$
$\log n$	$10^{100}n$	$n!$	$n \log n$
$n \cdot 2^n$	n^n	$\sqrt{\log n}$	n

Sample Solution

$$
\begin{array}{lllllllll}
 & \sqrt{\log n} & <_{\mathcal{O}} & \log(\sqrt{n}) & =_{\mathcal{O}} & \log n & =_{\mathcal{O}} & \log(n^2) \\
<_{\mathcal{O}} & (\log n)^2 & <_{\mathcal{O}} & \sqrt{n} & <_{\mathcal{O}} & n & =_{\mathcal{O}} & 10^{100}n \\
<_{\mathcal{O}} & n \log n & <_{\mathcal{O}} & n^2 & <_{\mathcal{O}} & n^{100} & <_{\mathcal{O}} & 2^n \\
<_{\mathcal{O}} & n \cdot 2^n & <_{\mathcal{O}} & 3^n & <_{\mathcal{O}} & n! & <_{\mathcal{O}} & n^n
\end{array}
$$