Exercise 1: Decidability

Let $\Sigma$ be a fixed finite alphabet. Show that the language of deterministic finite automata (DFAs) on $\Sigma$ 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.

Exercise 2: Landau Notation

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

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

For the following pairs of functions, check whether $f \in O(g)$ or $g \in O(f)$ 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) $f(n) = \log_2(2^n \cdot n^3)$, $g(n) = 3n$  \hspace{1cm} \textbf{Hint:} You may use that $\log_2 n \leq n$ for all $n \in \mathbb{N}$.

Exercise 3: Sorting Functions by Asymptotic Growth

Sort the following functions by asymptotic growth using the $O$-notation. Write $g \prec O f$ if $g \in O(f)$ and $f \notin O(g)$. Write $g =_O f$ if $f \in O(g)$ and $g \in O(f)$.

<table>
<thead>
<tr>
<th>Function</th>
<th>$n^2$</th>
<th>$\sqrt{n}$</th>
<th>$n^{100}$</th>
<th>$2^n$</th>
<th>$\log(n^2)$</th>
<th>$(\log n)^2$</th>
<th>$n!$</th>
<th>$\sqrt{\log n}$</th>
<th>$n \log n$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3^n$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\log n$</td>
<td>$n^{100}$</td>
<td>$10^{100}$</td>
<td>$n!$</td>
<td>$\sqrt{\log n}$</td>
<td>$n \log n$</td>
<td>$n$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>