University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour

Theoretical Computer Science - Bridging Course
Exercise Sheet 7

Due: Tuesday, 17th of June 2025, 12:00 pm

Exercise 1: The Halting Problem Revisited *(3+3 Points)*

Show that both the halting problem and its special version are both undecidable.

(a) The *halting problem* is defined as

$$H = \{\langle M, w\rangle \mid \langle M\rangle \text{ encodes a TM and } M \text{ halts on string } w\}.$$

Hint: Assume H is decidable and try to reach a contradiction by showing that some known undecidable problem (cf. from the lecture) is decidable.

(b) The *special halting problem* is defined as

$$H_s = \{\langle M\rangle \mid \langle M\rangle \text{ encodes a TM and } M \text{ halts on } \langle M\rangle\}.$$

Hint: Assume that M is a TM which decides H_s and then construct a TM which halts iff M does not halt. Use this construction to find a contradiction.

Exercise 2: A Non-Turing Recongnizable Problem *(3 Points)*

Fix an enumeration of all Turing machines (that have input alphabet Σ): $\langle M_1\rangle, \langle M_2\rangle, \langle M_3\rangle, \dots$.
Fix also an enumeration of all words over Σ: $w_1, w_2, w_3, \dots$.
Prove that language $L = \{w \in \Sigma^* \mid w = w_i, \text{ for some } i, \text{ and } M_i \text{ does not accept } w_i\}$ is not Turing recognizable.
Hint: Try to find a contradiction to the existence of a Turing machine that recognizes L.

Exercise 3: $\mathcal{O}$-Notation Formal Proofs *(1+2+2 Points)*

Roughly speaking, the set $\mathcal{O}(f)$ contains all functions that are not growing faster than the function f when additive or multiplicative constants are neglected. Formally:

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

For the following pairs of functions, state whether $f \in \mathcal{O}(g)$ or $g \in \mathcal{O}(f)$ or both. Proof 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 $n \in \mathbb{N}$.*

Exercise 4: Sort Functions by Asymptotic Growth *(6 Points)*

Give a sequence of the following functions sorted by asymptotic growth, i.e., for consecutive functions g, f in your sequence, it should hold $g \in \mathcal{O}(f)$. Write "$g \cong f$" if $f \in \mathcal{O}(g)$ *and* $g \in \mathcal{O}(f)$.

$\log_2(n!)$	$\sqrt{n}$	2^n	$\log_2(n^2)$
3^n	n^{100}	$\log_2(\sqrt{n})$	$(\log_2 n)^2$
$\log_{10} n$	$10^{100} \cdot n$	$n!$	$n \log_2 n$
$n \cdot 2^n$	n^n	$\sqrt{\log_2 n}$	n^2