University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
Z. Parsaeian

Theoretical Computer Science - Bridging Course
Sample Solution Exercise Sheet 7

Due: Tuesday, 10th of December 2024, 12:00 pm

Exercise 1: Undecidable or Not Turing recongnizable Problems *(4+4 Points)*

1. Show that $EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1, M_2$ are Turing Machines and $L(M_1) = L(M_2)\}$ is undecidable.

   Hint: You may use that $E_{TM} = \{\langle M\rangle \mid M$ is a Turing Machine and $L(M) = \emptyset\}$ is undecidable.

2. 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 $L = \{w \in \Sigma^* \mid w = w_i,$ for some $i,$ and M_i 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.

Sample Solution

1. Assume we had a TM R that decides EQ_{TM}. We construct a decider F for E_{TM} in the following and this will lead to a contradiction.
   $F=$ "On input $\langle M\rangle$ where M is a TM:

   - Construct a TM B that rejects all inputs.
   - Run R on $\langle M, B\rangle$. Accept iff R accepts."

2. Assume M is a turing machine recognizing L. Then there is an i such that $M = M_i$.

   Assume M accepts w_i. One the one hand this implies $w_i \in L$ (as M recognizes L), on the other hand it implies $w_i \notin L$ (by the definition of L), leading to a contradiction.

   Now assume M does not accept w_i. One the one hand this implies $w_i \notin L$ (as M recognizes L), on the other hand it implies $w_i \in L$ (by the definition of L), leading to a contradiction.

   So in either case we get a contradiciton. Therefore such a TM can not exist.

Exercise 2: The Halting Problem Revisited *(4+4 Points)*

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

1. 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.

2. 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.

Sample Solution

1. Assume H is decidable, hence there exists TM R that decides on it.
We know from the lecture that the A_{TM} problem is undecidable.
We reach a contradiction by constructing a TM S that decides on A_{TM} as follows.
$S=$ " On input $< M, w >$, where M is a TM and w is a string:
1. Run TM R on $< M, w >$, if R rejects, reject.
2. If R accepts, simulate M on w until it halts. If M accepts, accept; if M rejects, reject."

2. Assume that H_s is decidable. Then there is a TM M which decides it. Now let us define a TM $\tilde{M}$ as follows. TM $\tilde{M}$ on input w uses M to test whether $w \in H_s$. If $w \in H_s$ it enters a non terminating loop, otherwise it accepts w. We now apply $\tilde{M}$ on input $\langle\tilde{M}\rangle$ and construct a contradiction.

   $\langle\tilde{M}\rangle \notin H_s$: Then M rejects $\langle\tilde{M}\rangle$. Thus $\tilde{M}$ accepts $\langle\tilde{M}\rangle$ by the definition of $\tilde{M}$. Thus, $\langle\tilde{M}\rangle \in H_s$, a contradiction.

   $\langle\tilde{M}\rangle \in H_s$: Then M accepts $\langle\tilde{M}\rangle$, i.e., $\tilde{M}$ enters a non terminating loop on $\langle\tilde{M}\rangle$ and does not halt on $\langle\tilde{M}\rangle$ which means that $\langle\tilde{M}\rangle \notin H_s$, a contradiction.

$$\langle\tilde{M}\rangle \in H_s \Leftrightarrow \langle\tilde{M}\rangle \notin H_s$$

Exercise 3: $\mathcal{O}$-Notation Formal Proofs *(1+2+3 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 $n \in \mathbb{N}$.*

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 $n \geq M : \log_2(2^n \cdot n^3) \leq c \cdot n$. Indeed,

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

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

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 4: Sort Functions by Asymptotic Growth *(7 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

Sample Solution

For clarification, we write $g \lesssim f$ if $g \in \mathcal{O}(f)$, but not $f \in \mathcal{O}(g)$.

$$\begin{array}{lllllllll}
 & \sqrt{\log_2 n} & \lesssim & \log_2(\sqrt{n}) & \cong & \log_{10} n & \cong & \log_2(n^2) \\
\lesssim & (\log_2 n)^2 & \lesssim & \sqrt{n} & \lesssim & 10^{100} n & \lesssim & n \log_2 n \\
\cong & \log_2(n!) & \lesssim & n^2 & \lesssim & n^{100} & \lesssim & 2^n \\
\lesssim & n \cdot 2^n & \lesssim & 3^n & \lesssim & n! & \lesssim & n^n
\end{array}$$