Gruppe	Popov (8:30)	Popov (9:15)	Hemmecke (10:15)	Hemmecke (11:00)
Name		Matrikel		SKZ

Klausur 2

Berechenbarkeit und Komplexität

18. Januar 2019

Part 1 *RecFun2018*

Let $f : \mathbb{N} \to \mathbb{N}$ be a primitive recursive function and let $g : \mathbb{N} \to_p \mathbb{N}$ be a μ-recursive function.

1	yes	
2		no
3		no

1. *Is f necessarily μ-recursive?*
2. *If g is a total function, is it then also necessarily primitive recursive?*
3. *Assume that for every natural number x for which g is defined the equation $g(x) < f(x)$ holds. Can it be concluded that g is primitive recursive?*

Conterexample: Let g be a not total function.

4		no

Assume there is a LOOP program that for every natural number x computes the value $g(f(x))$. Is then g necessarily primitive recursive?

Counterexample: Let g be the (not primitive recursive) function that is only defined at 0, so that $g(0) = 0$ and let f be the zero function.

5		no

Assume there is a LOOP program that for every natural number x computes the value $f(g(x))$. Is then g necessarily primitive recursive?

Counterexample: f is the zero function and g is a total function that is not primitive recursive. Then $f \circ g = f$ and, therefore, primitive recursive, i.e., LOOP-computable.

Part 2 *Grammar2018*

Consider the grammar $G = (N, \Sigma, P, S)$ where $N = \{S, A\}$, $\Sigma = \{0, 1\}$, $P = \{S \to 1AA0, AA \to AAA, A \to \varepsilon\}$.

6		no	*Is $1000 \in L(G)$?*
7	yes		*Is $L(G)$ finite?*
8		no	*Is the grammar G context-free?*
9	yes		*Is there a right-linear grammar G' such that $L(G) = L(G')$?*
10	yes		*Does for every Turing machine M exist a grammar H such that $L(M) = L(H)$?*

Part 3 *Decidable2018*

Consider the following problems. In each problem below, the input of the problem is the code $\langle M \rangle$ of a Turing machine $M = (Q, \Gamma, \sqcup, \{0,1\}, \delta, q_0, F)$.
Let $L_A(M)$ be the set of words that M accepts in at most 2018 steps.
For $w \in \{0,1\}^*$ let $b(w)$ be the natural number denoted by the bitstring $1w$.
Problem A: *Is $L_A(M)$ finite?*
Problem B: *Is there a μ-recursive function f_M such that for every word $w \in \{0,1\}^*$ the Turing machine M halts on w if and only if $f_M(b(w)) = 0$?*
Problem C: *Does there exist a right-linear grammar G such that $L(M) = L(G)$?*
Problem D: *Does M halt on at least one word $w \notin L(M)$?*

11	yes	

Is A decidable?

If we find a word $w \in \{0,1\}^*$ of length exactly 2018 that is accepted by M, then any longer word with w as a prefix is also accepted, i.e., $L_A(M)$ is infinite. If such a word cannot be found, then since there is only a finite number of words with length < 2018, $L_A(M)$ is finite. In other words, A is decidable.

12	yes	

Is B decidable?

Every Turing machine M can be simulated by a WHILE program P_M. If M halts, also P_M halts. We can modify P_M to another WHILE program W_M such that it returns 0 in this case. If M does not halt, also P_M (and therefore W_M) does not halt. Clearly W_M computes a μ-recursive function f_M with the properties given in Problem B. In other words, such an f_M does exist for every Turing machine M. Problem B can trivially be answered by a Turing machine that always says "yes". Problem B is, therefore, decidable.

13		no

Is C decidable?

Rice Theorem

14	yes	

Is D semi-decidable?

Run M (in parallel) on all words (usual trick of doing one step of the run of all instances of M and starting a new instance of M on the next word). Whenever an instance halts in a non-accepting state, the answer to problem D is "yes".

15	yes	

Let $P \subseteq \{0,1\}^*$ be a decision problem such that the restricted Halting problem is is reducible to P. Can it be concluded that P is undecidable?

Part 4 *Complexity2018*

Let $f(n) = 20^n + n^{18}$, $g(n) = n^{20} + 18^n$, and $h(n) = n^{20} \cdot \log_2(n^{18})$.

16		no	*Is it true that $f(n) = \Theta(g(n))$?*
17		no	*Is it true that $g(n) = O(h(n))$?*
18		no	*Is it true that $100^n = O(10^n)$?*
19	yes		*Is it true that $n! = O(n^n)$?*

Part 5 *LoopWhile2018*

Let $f, g : \mathbb{N}^2 \to \mathbb{N}$ be defined as follows

$$f(a,b) := \begin{cases} 1, & \text{if } a < b, \\ 0, & \text{otherwise;} \end{cases} \qquad g(a,b) := \begin{cases} 0, & \text{if } a < b, \\ 1, & \text{otherwise.} \end{cases}$$

20	yes		*Are both f and g LOOP computable functions?*
21	yes		*Is (μf) a LOOP computable function?*

$(\mu f)(b) = b$.

22		no

Is (μg) a LOOP computable function?

$(\mu g)(0)$ is undefined.

23	yes	

Are both (μf) and (μg) WHILE computable functions?

Part 6 *OpenComputability2018*

The syntax of a LOOP program is given by:

$$P ::= x_i = 0 \mid x_i := x_j + 1 \mid x_i := x_j - 1 \mid P; P \mid \textbf{loop } x_i \textbf{ do } P \textbf{ end}$$

Please note that the arithmetic operations allowed in a LOOP program are only $x_i := x_j + 1$ and $x_i := x_j - 1$.

24 *1 Point* *Write a LOOP program that computes the function $c(n) = \sum_{k=1}^{n} k^2$.*

loop x_1 do // for $x_1 = n$ downto 1
    loop x_1 do // Compute $x_0 := x_0 + x_1^2$.
        loop x_1 do $x_0 := x_0 + 1$; end;
    end;
    $x_1 := x_1 - 1$;
end;

25 *1 Point* *Let $B(n)$ be the minimal number of commands of the form $x_i := x_j + 1$ that are executed by a LOOP program that computes $c(n)$. Express $B(n)$ in Ω notation.*
$B(n) = \Omega($ $)$

The result $c(n) = \frac{n(n+1)(2n+1)}{6}$ can only be achieved by executing $\Omega(n^3)$ times a command of the form $x_i := x_j + 1$.