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

Klausur 2

Berechenbarkeit und Komplexität

15-Januar-2016

Part 1 RecFun2015

Let f_1 and f_2 be two μ-recursive functions that are not primitive recursive. And let g be a primitive recursive function. All function are to be understood as mappings $\mathbb{N} \to_P \mathbb{N}$.

1	yes	

Is it possible that the composition $f_1 \circ f_2$ is primitive recursive?

If f_1 is a function that is undefined on the input 0 and returns 1 for every other input and f_2 is a total function that is not primitive recursive such that $f_2(n) > 0$, e.g. $f_2(n) = 1 + \operatorname{ack}(n, n)$. Then $(f_1 \circ f_2)(n) = 1$ and this function is surely primitive recursive.

2	yes	

Is g μ-recursive?

Every primitive recursive function is μ-recursive.

3	yes	

Let $f : \mathbb{N} \to_P \mathbb{N}$ be a partial function that is undefined for every input. Is f a μ-recursive function?

Let $g : \mathbb{N}^2 \to \mathbb{N}$ be any primitive recursive function and let $h(x, y) = s(g(x, y))$. Then $f(y) := (\mu h)(y)$ is clearly a μ-recursive function with that is nowhere defined.

4	yes	

Is there a Turing machine that (on input n) returns $f_1(n)$ if $g(n)$ is odd and returns $f_2(n)$ if $g(n)$ is even?

5		no

Can every μ-recursive function be computed by a LOOP program?

Part 2 Grammar2015

Consider the grammar $G = (N, \Sigma, P, S)$ where $N = \{S\}$, $\Sigma = \{a, b\}$, $P = \{S \to aSB, S \to bB, B \to b\}$.

6	yes	

Is $aabbbb \in L(G)$?

7		no

Can $L(G)$ be generated by a nondeterministic finite state machine?

$L(G) = \left\{\, a^n b^{n+2} \,\middle|\, n \in \mathbb{N} \setminus \{0\} \,\right\}$. Pumping Lemma.

8	yes	

Does for every right-linear grammar R with terminal symbols $\Sigma = \{0, 1\}$ exist a Turing machine M such that $L(R) = L(M)$.

By Theorem 22 (lecture notes) is the language of a right linear grammar a regular language.

9		no

Does there exist a recursively enumerable language that can not be generated by a grammar?

Part 3 LoopWhile2015

Let a function $f : \mathbb{N}^2 \to \{0, 1, 2\}$ be defined by

$$f(a, b) := \begin{cases} 0 & \text{if } a = b, \\ 1 & \text{if } a < b, \\ 2 & \text{if } a > b. \end{cases}$$

10	yes		*Is f a LOOP computable function?*
11	yes		*Let f' be defined like f, but with the exception that f' is undefined if the first or the second argument is equal to 2016. Is f' a WHILE computable function?*
12	yes		*Is (μf) computable by a Turing machine?*
13	yes		*Let P be a WHILE program that computes a (partial) function $g : \mathbb{N}^2 \to_P \mathbb{N}$. Is g a μ-recursive function?*

Part 4 *Complexity2015*

Let $f(n) = n \log_2(n)$, $g(n) = n2^n$, $h(n) = \log_2(n)2^n$.

14		no	*Is it true that $f(n) + g(n) = O(f(n))$?*
15		no	*Is it true that $2^{f(n)} = O(h(n))$?*
16		no	*Is it true that $h(n)^2 = O(g(n))$?*
17	yes		*Is it true that $f(n)g(n) = O(h(n)^2)$?*
18	yes		*Is it true that $\frac{n^4+n+1}{n^2+1} = O(n^2 \log_2(n))$?*

Part 5 *Decidable2015*

Consider the following problems. In each problem below, the input of the problem is the code $\langle M \rangle$ of a Turing machine M with input alphabet $\{0, 1\}$.

Problem F: *Does $L(M)$ contain exactly 2016 words?*

Problem W: *Does $L(M)$ contain at least 2016 words?*

Problem H: *Does M halt after at most 2016 steps?*

19		no	*Is F decidable?*

Rice theorem

20	yes		*Is W semi-decidable?*

Generate all words from $\{0, 1\}^*$. Simulate M on each word. One has to start a separate simulation for each word and run them all in parallel. If at least 2016 of these simulations stop with acceping the respective input word then the answer to the problem is YES.

21	yes		*Is H decidable?*

We only have to investigate 2016 steps of M. Thus, M can at least have touched 2016 cells on the tape. So we simulate (at most) 2016 steps of M on each possible word of length 2016. If M stops in each of these finitely many cases then the answer to the problem is YES, otherwise it is NO. So the problem is decidable.

22		no	*Suppose a problem $P \subseteq \{0, 1\}^*$ is decidable. Is then also every subproblem $U \subseteq P$ decidable?*

Counter example. Let $P = \{0, 1\}^*$. This is surely decidable, since it is a recursive language. There are, however subsets of $\{0, 1\}^*$ that are not recursive languages and thus also not decidable.

23	yes		*Let $P, P' \subseteq \{0, 1\}^*$ and let M be a Turing machine that for every $w \in P$ computes a $w' \in P'$ and for every $w \notin P$ computes a word $w' \notin P'$. Assume P' is decidable. Can it be concluded that P is semi-decidable.*

By Theorem 32 (lecture notes) we can even conclude that P is decidable. Note that the "computable function" that is required in Definition 42 is the function computed by M.

Part 6 $\boxed{\textit{OpenComputability2015}}$

Let $T(n)$ be the number of multiplications executed during the run of the following program while evaluating $g(n,1)$.

```
function g(n, x) {
    if n==0
        return x
    else
        k = floor(n/3)  //floor(x) = biggest integer less than or equal to x
        return g(k, x) * g(k, x+1)
```

24 | *1 Point*

Compute $T(9)$.

$T(9) =$

$g(9,1) = g(3,1) * g(3,2) = g(1,1) * g(1,2) * g(1,2) * g(1,3) =$
$g(0,1) * g(0,2) * g(0,2) * g(0,3) * g(0,2) * g(0,3) * g(0,3) * g(0,4) =$
$1 * 2 * 2 * 3 * 2 * 3 * 3 * 4$. So $T(9) = 7$.

25 | *1 Point*

Determine $T(n)$ asymptotically for large n. Use Θ-notation.

$T(n) =$

A recursion formula for T is $T(n) = 1 + 2T(\lfloor \frac{n}{3} \rfloor)$. By the Master Theorem we get $T(n) = \Theta(n^{\log_3 2}) = O(2^{\log_3 n})$