Problems Solved:	36	37	38	39	40

Name:

Matrikel-Nr.:

Problem 36. Let

$$Y := (\lambda f.((\lambda x.(f(xx)))(\lambda x.(f(xx))))).$$

(just as in Section 3.2 of the Lecture Notes). Show by an explicit derivation that

$$(YF) \to^* (F(YF)).$$

Problem 37. True or false?

1. $(2n + 3)(3n + 2) = O(n^2)$
2. $(2n + 3) + \log_2(3n^6 + 2) = O(n)$
3. $\frac{1024}{2^n} = O(1)$
4. $\frac{1024}{2^n} = \Theta(\log_2(n))$
5. $4^n = O(2^n)$
6. $2^n = O(4^n)$

Prove your answers based on Definition 45 from the lecture notes.

Problem 38. Let $f, g, h : \mathbb{N} \to \mathbb{R}_{\geq 0}$. Prove or disprove based on Definition 45 from the lecture notes.

1. $f(n) = O(f(n))$
2. $f(n) = O(g(n)) \implies g(n) = O(f(n))$
3. $f(n) = O(g(n)) \wedge g(n) = O(h(n)) \implies f(n) = O(h(n))$

Problem 39. Write a LOOP program that computes the function $f : \mathbb{N} \to \mathbb{N}$, $f(n) = 2^n$.

1. Count the number of variable assignments (depending on n) during the execution of your LOOP program with input n.
2. What is the time complexity of your program (depending on n)?
3. Is it possible to write a LOOP program with time complexity better than $O(2^n)$? Give an informal reasoning of your answer.
4. Let $l(k)$ denote the bit length of a number $k \in \mathbb{N}$. Let $b = l(n)$, i.e., b denotes the bit length of the input. What is the time complexity of your program depending on b, if every variable assignment $x_i := x_j + 1$ costs time $O(l(x_j))$?

Problem 40. Let $\Sigma = \{0, 1\}$ and let $L \subseteq \Sigma^*$ be the set of binary numbers divisible by 3, i.e.,

$$L = \{x_n \ldots x_1 x_0 \ : \ 3 \text{ divides } \sum_{k=0}^{n} x_k 2^k\}.$$

(By convention, the empty string ε denotes the number 0 and so it is in L too.)

1. Design a Turing machine M with input alphabet Σ which recognizes L, halts on every input, and has (worst-case) time complexity $T(n) = n$. Write down your machine formally. (A picture is not needed.) *Hint:* Three states q_0, q_1, q_2 suffice. The machine is in state q_r if the bits read so far yield a binary number which leaves a remainder of r upon division by 3. The transition from one state to another represents a multiplication by 2 and the addition of 0 or 1.
2. Determine $S(n)$, $\bar{T}(n)$ and $\bar{S}(n)$ for your Turing machine.
3. Is there some faster Turing machine that achieves $\bar{T}(n) < n$? (Justify your answer.)