Universität Koblenz-Landau

FB 4 Informatik

Prof. Dr. Viorica Sofronie-Stokkermans
Dipl. Inform. Markus Bender

December 12, 2013

Exercises for
"Advances in Theoretical Computer Science"
Exercise sheet 8

Exercise 8.1:

Prove that the following functions are primitive recursive:

(1) $\mathsf{lcm} : \mathbb{N}^2 \rightarrow \mathbb{N}$ defined by $\mathsf{lcm}(n, m) = \begin{cases} 0 & \text{if } n = 0 \text{ or } m = 0 \\ k & \text{if } n \neq 0, m \neq 0 \text{ and} \\ & k \text{ is the least common multiple of } m \text{ and } n \end{cases}$

(2) $\mathsf{gcd} : \mathbb{N}^2 \rightarrow \mathbb{N}$ defined by $\mathsf{gcd}(n, m) = \begin{cases} 0 & \text{if } n = 0 \text{ or } m = 0 \\ k & \text{if } n \neq 0, m \neq 0 \text{ and} \\ & k \text{ is the greatest common divisor of } n \text{ and } m \end{cases}$

Remark: You are allowed to use all functions that were proved to be primitive recursive in the lecture or in a previous exercise.

Exercise 8.2:

Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be defined as follows:

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 2 & \text{if } n = 1 \\ 4 & \text{if } n = 2 \\ f(n-1) + f(n-2) + f(n-3) & \text{if } n > 2 \end{cases}$$

Prove that f is primitive recursive.

Remark: You are allowed to use all functions that were proved to be primitive recursive in the lecture or in a previous exercise.

Exercise 8.3:

In order to prove that $\mathcal{P} \subseteq$ LOOP in the lecture from 12.12.2013 it was shown that:

- all atomic primitive recursive functions are LOOP computable
- LOOP is closed under composition of functions
- LOOP is closed under primitive recursion

Use induction on the structure of primitive recursive functions to show (with the help of this result) that all primitive recursive functions are LOOP computable.

Exercise 8.4:
Show that there exists a function $f : \mathbb{N} \to \mathbb{N}$ which is not primitive recursive.

Hint: Since $\mathcal{P}$ = LOOP and we showed that the set of LOOP programs is recursively enumerable it follows that $\mathcal{P}$ is recursively enumerable.

Consider an enumeration of all (unary) primitive recursive functions $f_1, f_2, \ldots$. In order to construct a function which is not primitive recursive you can for instance use an idea similar to that used in the proof of the fact that LOOP $\neq$ TM (see the slides from 21.11.2012, page 50).

The submission of the solutions is not compulsory. If you want to submit your solutions, please do so until Tuesday, 17.12.2013, 10:00 s.t.. Joint solutions prepared by up to three persons are allowed. Please do not forget to write your name on your solution.
Submission possibilities:

- By e-mail to mbender@uni-koblenz.de with the keyword "Homework ACTCS" in the subject.
- Put it in the box in front of Room B 222.