Problems Solved:	26	27	28	29	30

Name:

Matrikel-Nr.:

Problem 26. Let m be the function defined by

$$m(x, y) = \begin{cases} 0 & \text{if } x < y, \\ x - y & \text{otherwise.} \end{cases}$$

Show that m is indeed a primitive recursive function by defining it explicitly from the base functions, composition, and the primitive recursion scheme.
Note that according to Definition 29 (lecture notes), in the composition scheme, the g_i have the same number of arguments as h. Similarly, in the primitive recursion scheme, recursion is done on the first argument of h and the respective f has one argument less while g has one argument more than h.
For your solution, you are not allowed to deviate from these formal requirements.
Hint: Define first a helper function "pred" that behaves like $m(x, 1)$, but has only one argument. Then note that for $y > 0$ we have $m(x, y) = pred(m(x, y - 1))$, i.e, the recursion would happen in the second argument (which is not allowed according to Definition 29). However, one can easily exchange arguments by means of the projection function.

Problem 27. Let $q : \mathbb{N}^2 \to \mathbb{N}$, $(x, y) \mapsto x \cdot x$ (sic!) and $u : \mathbb{N}^2 \to \mathbb{N}$,

$$u(x, y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y, \end{cases}$$

be given primitive recursive functions.
Let $r : \mathbb{N}^2 \to \mathbb{N}$ be definied by

$$\begin{aligned} r(x) &= (\mu p)(x) && \text{minimization} \\ p(y, x) &= u(q(y, x), \mathrm{proj}_2^2(y, x)) && \text{composition} \end{aligned}$$

Informally we have

$$r(x) = \min_y \{ y \in \mathbb{N} \mid u(q(y, x), x)) = 0 \}$$

Similar to the treatise in the lecture notes, construct a while program that computes r. For simplicity, you are allowed to write statements such as $x_k = q(x_i, x_j)$ and $x_k = u(x_i, x_j)$ into your program. What does $r(x)$ return? What will your program compute if it is started with input $x_1 = 2$?

Problem 28. Consider the following term rewriting system:

$$p(x, s(y)) \to p(s(x), y) \tag{1}$$

$$p(x, 0) \to x \tag{2}$$

1. Show that
$$p(s(0), s(0)) \xrightarrow{*} s(s(0))$$
by a suitable reduction sequence. For each reduction step, underline the subterm that you reduce, and indicate the reduction rule and the matching substitution σ used explicitly.

2. Disprove that
$$p(p(s(0), s(0)), p(s(0), s(0))) \xrightarrow{*} s(s(0)).$$

Problem 29. According to Definition 32 of the lecture notes, there are no natural numbers in Lambda calculus. However, natural numbers can be encoded (known as Church encoding) as "Church numerals" (see below), i.e., as functions $\mathbf{n}$ that map any function f to its n-fold application $f^n = f \circ \ldots \circ f$. Note that we denote such a "natural number" representation via boldface symbols in order to emphasize that these are lambda terms. In other words, we define Church numerals as follows. By letting "application" bind stronger than "abstraction", we avoid writing parentheses where appropriate.

$$\begin{aligned}
\mathbf{0} &= \lambda f.\lambda x.x \\
\mathbf{1} &= \lambda f.\lambda x.fx \\
\mathbf{2} &= \lambda f.\lambda x.f(fx) \\
\mathbf{3} &= \lambda f.\lambda x.f(f(fx)) \\
\mathbf{4} &= \lambda f.\lambda x.f(f(f(fx))) \\
&\vdots \\
\mathbf{n} &= \lambda f.\lambda x.\underbrace{f(\cdots(f}_{n\text{-fold}}\,x)\cdots)
\end{aligned}$$

1. Define a lambda term add that represents addition of "Church numerals".

2. Show the intermediate steps of a reduction from ((add $\mathbf{2}$) $\mathbf{1}$) to $\mathbf{3}$.

Hint: a bit of literature research may help.

Problem 30. Consider the grammar $G = (N, \Sigma, P, S)$ where $N = \{S\}$, $\Sigma = \{a, b, c, d\}$, $P = \{S \to a, S \to b, S \to dScSd\}$.

(a) Is $daacbd \in L(G)$?

(b) Is $dddacadcbdcbd \in L(G)$?

(c) Does every element of $L(G)$ contain an even number of occurrences of d?

(d) Is $L(G)$ regular?

(e) Is $L(G)$ recursive?

Justify your answers.