Homework 1 for the course Computability Theory

Deadline: April 12, 2021

Note: For all Turing machines, register machines, and **PRL**-programs that you define, you shall also include informal comments clearly describing what the program is supposed to do.

1. Prove that the following functions, predicates, and sets are primitive recursive:
   - $f(x) = x!$ (where $0! = 1$).
   - $\overline{\text{sg}}(x) = \begin{cases} 0 & x > 0, \\ 1 & x = 0. \end{cases}$
   - $\text{Rem}(x, y) = \begin{cases} \text{the reminder of dividing } y \text{ by } x & \text{if } x \neq 0 \\ 0 & \text{otherwise.} \end{cases}$
   - $f(x) = F_x$, where F_x is the xth Fibonacci number.
   - The union of any two co-finite sets.
2. Describe the function which is the result of application of the minimization operator to $g(x, y, z) = |zy - x|$.
3. Let $f(x, y) = x \dot{-} y$. Write a **PRL**-program P such that $Func_{2,1}(P, x)$ computes f.
4. Write a program for the Turing machine computing the partial function $f(x, y) = x - 2y + 3$.
5. Prove that if $f(x) = \mu y[g(x, y) = 0]$ and $g(x, y)$ is register-computable, then $f(x)$ is also register-computable.
6. Recall Definition 27 from the script:

Definition 27. We define

$$
\begin{aligned}
f_0(0) &= 1, \\
f_0(1) &= 2, \\
f_0(x) &= x+2, \text{ for } x > 1. \\
\\
f_{i+1}(x) &= f_i^x(1), \text{ for all } i, x.
\end{aligned}
$$

Here f_i^k denotes k iterations of f_i: $f_i^k(x) = f_i(f_i(\ldots(f_i(x))\ldots))$.

Prove the following properties of the functions $f_i, i \in \omega$ (Lemma 7 in the script):

Lemma 7. For all k, x we have

a) $f_i(x) \geq x + 1$ for all i,

b) $f_i^k(x)$ is non-decreasing in i, k, x,

c) $2f_i^k(x) \leq f_i^{k+1}(x)$ for $i \geq 1$.

d) $f_i^k(x) + x \leq f_i^{k+1}(x)$, for $i \geq 1$.