# Theory of Computation Excercise for TTF (5)

Dimitri Tabatadze · Saturday 30-03-2024

## Problem 5.1:

Function $h : \mathbb{N}^2 \to \mathbb{N}$ which enumerates tuples of natural numbers can be written as

$$h(a, b) = \frac{(a + b - 2)(a + b - 1)}{2} + b.$$

a) Let $f : A \to \mathbb{N}$ and $g : B \to \mathbb{N}$ be enumerations of $A$ and $B$. We can now construct a function $e : A \times B \to \mathbb{N}$ which enumerates $A \times B$ as follows:

$$e(a, b) = h(f(a), g(b))$$

b) We can enumerate $\mathbb{N}^k$ as follows:

$$f_1(n) = n$$
$$f_k(n_1, n_2, ..., n_k) = h(n_1, f_{k-1}(n_2, n_3, ..., n_k))$$

and for $\mathbb{N}_0^k$ we can

$$g_1(n) = n + 1$$
$$g_k(n_1, n_2, ..., n_k) = h(n_1 + 1, g_{k-1}(n_2, n_3, ..., n_k))$$

so $\mathbb{N}_0^k$ is enumerable.

c) A subset of natural numbers $U$ can be represented by a sequence $\{a_i\}_{i\in\mathbb{N}}$ with $a_i = \begin{cases} 1 & i\in U \\ 0 & i\notin U \end{cases}$ so he set of all subsets of natural numbers can be written as a sequence $\{S_k\}_{k\in\mathbb{N}}$ of such sequences. Now we can employ diagonalization to show that there will exist a sequence $D$ such that $\nexists k \in \mathbb{N} : S_k = D$. To do this we just define $D$ as $D_i = (S_i)_i \oplus 1$. This way $D$ will be different from every $S_k$ by at least one element.

d) Similarly to c), assume $\{a_i\}_{i\in\mathbb{N}}$ is the sequence of digits of a real number. Let $\{S_k\}_{k\in\mathbb{N}}$ be the sequence of all such real numbers. Construct $\{d_i\}_{i\in\mathbb{N}}$ such that $d_i \neq (S_i)_i$. This way $D$ will be different from every $S_k$ by at least one element.

## Problem 5.2:

a) Let

$$\min(x, 0) = 0$$
$$\min(0, y) = 0$$
$$\min(x + 1, y + 1) = s(\min(x, y))$$

and

$$\max(x, 0) = x$$
$$\max(0, y) = y$$
$$\min(x + 1, y + 1) = s(\max(x, y))$$

b) Let $\text{abs}(x, y) = |x - y|$

$$\text{abs}(x, y) = \text{sub}(\max(x, y), \min(x, y))$$

now we need to define $\text{sub}(x, y)$

$$\text{sub}(x, 0) = x$$
$$\text{sub}(x, y + 1) = p(\text{sub}(x, y))$$

where

$$p(0) = 0$$
$$p(x+1) = x.$$

## Problem 5.3:

- $$\begin{aligned} q(x_0, \dots, x_{k-1}, y) &= S(p_k(x_0, \dots, x_{k-1}, y)) \\ h(x_0, \dots, x_{k-1}, y) &= f(p_1(x_0, \dots, x_{k-1}, y), \dots, p_{k-1}(x_0, \dots, x_{k-1}, y), q(x_0, \dots, x_{k-1}, y)) \\ g(x_0, \dots, x_{k-1}, y) &= \mathrm{add}(p_k(x_0, \dots, x_{k-1}, y), h(x_0, \dots, x_{k-1}, y)) \\ \mathrm{bsum}(x_0, \dots, x_{k-1}, 0) &= f(x_0, \dots, x_{k-1}, 0) \\ \mathrm{bsum}(x_0, \dots, x_{k-1}, y+1) &= g(x_0, \dots, x_{k-1}, \mathrm{bsum}(x_0, \dots, x_{k-1}, y), y) \end{aligned}$$
- $$\begin{aligned} g(x_0, \dots, x_{k-1}, ) \\ \mathrm{bprod}(x_0, \dots, x_{k-1}, 0) &= c_0(x_0, \dots, x_{k-1}) = 0 \\ \mathrm{bprod}(x_0, \dots, x_{k-1}, y+1) &= \mathrm{mult}(f(x_0, \dots, x_{k-1}, y+1), \mathrm{bprod}(x_0, \dots, x_{k-1}, y)) \end{aligned}$$

## Problem 5.4:

First define a new successor-like function

$$f(z, y, x) = S(p_3(z, y, x)).$$

Now we can replace the successor in the recursion step with $f$

$$\mathrm{add}(0, x) = p_1(x) = x$$
$$\mathrm{add}(y+1, x) = f(x, y, \mathrm{add}(x, y))$$

Same here, just define a *wrapper* for add

$$g(z, y, x) = \mathrm{add}(p_3(z, y, x), p_2(z, y, x)).$$

and replace add with $g$ in the recursion step.

$$\mathrm{mult}(x, 0) = c_0(x) = 0$$
$$\mathrm{mult}(x, y+1) = g(y, x, \mathrm{mult}(x, y))$$