Jméno: Ferdo Mravec **UČO:** 1234567

0007 *list* 1 *učo* 1234567 *body*

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0 1 2 3 4 5 6 7 8 9

2. [3 body] Pro každé $j \in \mathbb{N}$ definujeme množinu $A_j \subseteq \mathbb{N}$ následujícím způsobem:

$$A_j = \bigcup_{l \in range(\varphi_j)} range(\varphi_l)$$

(a) Dokažte, že existují indexy $j, k \in \mathbb{N}$ takové, že A_j je rekurzivní a A_k není rekurzivní.

(b) Rozhodněte a dokažte, zda jsou všechny množiny A_j rekurzivně spočetná.

(a) Nechť j je index prázdné funkce, tedy $\varphi_j(x) = \bot$. Jelikož $range(\varphi_j) = \emptyset$ a sjednocení přes prázdnou množinu je prázdná množina, dostáváme $A_j = \emptyset$, což je rekurzivní množina.

Zbývá najít k takové, že A_k není rekurzivní. Jelikož problém zastavení K je rekurzivně spočetná množina, existuje vyčíslitelná funkce φ_m splňující $range(\varphi_m) = K$. Nechť k je index konstantní funkce m, tedy $\varphi_k(x) = m$. Platí $range(\varphi_k) = \{m\}$ a celkově dostáváme

$$A_k = \bigcup_{l \in range(\varphi_k)} range(\varphi_l) = \bigcup_{l \in \{m\}} range(\varphi_l) = range(\varphi_m) = K$$

Množina $A_k = K$ není rekurzivní.

(b) Ukážeme, že každá množina A_j je rekurzivně spočetná. Pro libovolné pevné $j \in \mathbb{N}$ zkonstruujeme program, který generuje množinu A_j pomocí příkazu $output(x)$. Nejprve přepíšeme definici A_j do ekvivalentního tvaru

$$\begin{aligned} A_j &= \{i \mid \exists k \,.\, k \in dom(\varphi_j) \text{ a } i \in range(\varphi_{\varphi_j(k)})\} \\ &= \{\varphi_{\varphi_j(k)}(y) \mid \exists k, y \,.\, k \in dom(\varphi_j) \text{ a } y \in dom(\varphi_{\varphi_j(k)})\}. \end{aligned}$$

Nyní snadno napíšeme program, který prochází všechny trojice $(k, y, z) \in \mathbb{N}^3$ a funkcí Sc zjišťuje, zda výpočty $\varphi_j(k)$ a $\varphi_{\varphi_j(k)}(y)$ pomocí příslušných programů zastaví během z kroků. Pokud tomu tak je, program dá na výstup hodnotu $\varphi_{\varphi_j(k)}(y)$. Připomínáme, že j reprezentuje v programu fixní hodnotu a nikoliv promněnnou.

```
begin
  n := 0;
  while true do begin
      k := π1(n);
      y := π1(π2(n));
      z := π2(π2(n));
      if Sc(j, k, z) = 1 then begin
          l := Φ(j, k);
          if Sc(l, y, z) = 1 then output(Φ(l, y))
        end;
      n := n + 1
    end
end
```

Dodejme, že pokud $k \in dom(\varphi_j)$ a $y \in dom(\varphi_{\varphi_j(k)})$, tak musí existovat z splňující $Sc(j, k, z) = 1$ a $Sc(l, y, z) = 1$, kde $l = \varphi_j(k)$. Tedy program generuje právě množinu A_j a ta je tudíž rekurzivně spočetná.