IB107 úkol 2, příklad 2 **Odevzdání: 2. 11. 2021 23:59**

Jméno: Brouk Pytlík **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. 0123456789

2. [3 body] Pro libovolnou funkci $f : \mathbb{N} \to \mathbb{N}$ definujeme binární relaci $R_f \subseteq \mathbb{N}^2$ vztahem

$$R_f = \{(x, f(x)) \mid x \in dom(f)\}.$$

Dokažte nebo vyvraťte každé z následujících tvrzení.

(a) f je vyčíslitelná funkce $\Longrightarrow$ relace R_f je rekurzivní

(b) f je vyčíslitelná funkce $\Longrightarrow$ relace R_f je r.e.

(c) f je vyčíslitelná funkce $\Longleftarrow$ relace R_f je r.e.

V důkazu můžete využít fakt, že následující funkce je totálně vyčíslitelná.

$$Sc'(x, y_1, y_2, z) = \begin{cases} 1 & \text{pokud program } P_x \text{ na vstupu } (y_1, y_2) \text{ zastaví během } z \text{ kroků} \\ 0 & \text{jinak} \end{cases}$$

(a) Tvrzení neplatí. Uvažme například funkci f definovanou takto:

$$f(x) = \begin{cases} 0 & \text{ak } \varphi_x(x) \neq \bot, \\ \bot & \text{inak.} \end{cases}$$

Tato funkce je zjevně vyčíslitelná (program simuluje pomocí univerzální funkcie výpočet $\varphi_x(x)$ a pokud výpočet skončí, vrací hodnotu 0). Pro funkci f platí

$$R_f = \{(x, 0) \mid \varphi_x(x) \neq \bot\}.$$

Vidíme, že druhým řezem relace R_f pro hodnotu 0 je problém zastavení $K = \{x \mid \varphi_x(x) \neq \bot\}$. Jelikož libovolný řez rekurzivní množiny je opět rekurzivní množina, relace R_f není rekurzivní, protože pak by musela být rekurzivní také množina K, což není.

(b) Tvrzení platí. Nechť f je libovolná vyčíslitelná funkce a i_f je nějaký její index. Zkonstruujeme vyčíslitelnou funkci $g : \mathbb{N}^2 \to \mathbb{N}$ splňující $dom(g) = R_f$, čímž dokážeme, že R_f je rekurzivně spočetná. Funkce g je počítaná nasledujícím programem.

begin
  $x_3 := \Phi(i_f, x_1)$;
  while $x_3 \neq x_2$ **do** **begin** **end**
end

Pokud vstup (x_1, x_2) patří do R_f, tak výpočet $\Phi(i_f, x_1)$ skončí a zároveň bude platiť $x_3 = x_2$, tedy i náš program skončí a $g(x_1, x_2)$ je definováno. Pokud vstup (x_1, x_2) do R_f nepatří, mohou nastat dvě možnosti. Jestliže $f(x_1)$ není definováno, tak výpočet $\Phi(i_f, x_1)$ cyklí a $g(x_1, x_2)$ není definováno. Jestliže $f(x_1)$ je definováno, ale $x_2 \neq f(x_1)$, tak výpočet $\Phi(i_f, x_1)$ skončí, ale náš program cyklí, protože nikdy neopustí *while*-cyklus. Tedy opět $g(x_1, x_2)$ není definováno. Celkem dostáváme $dom(g) = R_f$.

Jméno: Brouk Pytlík **UČO:** 1234567

0007 *list* 2 *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

(c) Tvrzení platí. Nechť R_f je relace definovaná pro nějakou funkci $f : \mathbb{N} \to \mathbb{N}$ dle zadání. Pokud je R_f rekurzivně spočetná, pak existuje vyčíslitelná funkce $g : \mathbb{N}^2 \to \mathbb{N}$ splňující $dom(g) = R_f$. Nechť i_g je nějaký index této funkce g. Následující program počítá funkci f.

```
begin
   counter := 0;
   done := 0;
   while done = 0 do
         begin
            result := π1(counter);
            steps := π2(counter);
            done := Sc'(ig, x1, result, steps);
            counter := counter + 1
         end
   x1 := result
end
```

Program využívá makra implementující totálně vyčíslitelnou funkci Sc' ze zadání a totálně vyčíslitelné projekční funkce π_1, π_2 z přednášky.

Uvažme libovolné pevné x_1. Není-li $f(x_1)$ definováno, v relaci R_f není žádná dvojice (x_1, result), a proto $g(x_1, \mathit{result}) = \perp$ a $Sc'(i_g, x_1, \mathit{result}, \mathit{steps}) = 0$ pro libovolné hodnoty *result* a *steps*. Program tedy cyklí, což odpovídá předpokladu $f(x_1) = \perp$. Je-li naopak $f(x_1)$ definováno, existuje právě jedna hodnota $\mathit{result} = f(x_1)$ splňující $(x_1, \mathit{result}) \in R_f$. Pro tuto hodnotu *result* musí být $g(x_1, \mathit{result})$ definováno, a tedy $Sc'(i_g, x_1, \mathit{result}, \mathit{steps}) = 1$ pro dostatečně velkou hodnotu *steps*. Postupným zvyšováním hodnoty *counter* prohledáváme všechny dvojice hodnot *result* a *steps* a v konečném čase nějakou dvojici splňující $Sc'(i_g, x_1, \mathit{result}, \mathit{steps}) = 1$ objevíme. Náš program v tom případě skončí s hodnotou *result* na výstupu. Program tedy implementuje funkci f a ta je proto vyčíslitelná.