Cvičení ke kursu *Vyčíslitelnost*, část I
(7. dubna 2012)

1. Nechť A a B jsou libovolné množiny, $f : A \to B$ a nechť $X \subseteq A$ a $Y \subseteq B$. *Obraz* množiny X přes funkci f je množina
$$f''X = \{\, y \in B\,;\ \exists x \in X(f(x) = y) \,\}$$
a *vzor* množiny Y přes funkci f je množina
$$f^{-1}{}''Y = \{\, x \in A\,;\ f(x) \in Y \,\}.$$
(a) Zdůvodněte, že $f^{-1}{}''Y$ je oprávněné značení pro vzor množiny Y přes funkci f: definici obrazu přes funkci lze přímočaře zobecnit na definici obrazu přes relaci, a vzor množiny je totéž co obraz přes inverzní relaci (která nemusí být funkcí).
(b) Rozhodněte, zda pro každou funkci $f : A \to B$ a každé dvě množiny X a Y platí $f^{-1}{}''(f''X) = X$ a $f''(f^{-1}{}''Y) = Y$. Pokud ne, určete v obou případech, zda platí alespoň některá inkluze.

2. • Nechť pro funkce f a h z N do N a pro číslo a platí
$$f(0) = a, \qquad \forall x(f(x+1) = h(f(x))).$$
Je funkce h určena funkcí f jednoznačně?

3. Dokažte, že násobení a umocňování jako funkce dvou proměnných jsou primitivně rekurzívní. V obou případech řekněte přesně, ze kterých funkcí je daná funkce odvozena primitivní rekurzí, kolik mají proměnných a proč jsou primitivně rekurzívní. Použijte fakt, že sčítání dvou proměnných je primitivně rekurzívní, tuto funkci neodvozujte.

4. (a) Dokažte, že funkci $[x, y, z] \mapsto x \cdot y \cdot z$ lze odvodit z násobení dvou proměnných a ze základních funkcí bez užití operace primitivní rekurze a minimalizace.
(b) Nechť pro funkce f a h platí $f(0) = a$, $\forall x(f(x+1) = h(f(x), x))$. Dokažte, že je-li h primitivně rekurzívní, pak i f je primitivně rekurzívní.

   Návod. Nejprve přidejte jednu jalovou proměnnou, pak použijte primitivní rekurzi v povoleném tvaru a nakonec odstraňte nadbytečnou proměnnou dosazením konstanty.

5. Je-li A rekurzívní množina dvojic, pak i funkce ψ definovaná předpisem
$$\psi(x) = \begin{cases} \min\{\, y\,;\ A(x, y) \,\} & \text{když } \exists y A(x, y) \\ \uparrow & \text{jinak,} \end{cases}$$
kde $\uparrow$ znamená “nedefinováno”, je částečně rekurzívní. Dokažte.

6. Nechť χ a ψ jsou částečné funkce jedné proměnné. Rozhodněte, zda platí:
(a) Když $\chi \circ \psi$ je totální, pak i ψ je totální.
(b) Když $\chi \circ \psi$ je totální, pak i χ je totální.

7. • Dokažte, že funkce mod, která každé dvojici $[x, y]$ přiřazuje zbytek po dělení čísla x číslem y, je primitivně rekurzívní funkce. Zbytek po dělení číslem 0 definujte libovolně.

8. • Dokažte, že funkce div, která každé dvojici $[x, y]$ přiřazuje výsledek celočíselného dělení čísla x číslem y, je primitivně rekurzívní funkce.

   Návod. Odvoďte funkci div primitivní rekurzí, použijte funkci mod z cvičení 7.

9. Totální funkce $f : \mathrm{N} \to \mathrm{N}$ je obecně rekurzívní, právě když f chápaná jako množina dvojic, tj. množina $\mathrm{Graf}(f) = \{\,[x, y]\,;\, f(x) = y\,\}$, je obecně rekurzívní. Každá primitivně rekurzívní funkce má primitivně rekurzívní graf. Dokažte.

   Návod. V $\Leftarrow$ první části odvoďte f pomocí operace minimalizace z charakteristické funkce (dvou proměnných) množiny $\mathrm{Graf}(f)$. Lze použít verzi minimalizace ze cvičení 5.

10. • Jsou-li g a h dvě primitivně rekurzívní funkce, pak i funkce f, která má v bodech 0, 1, 2, 3, 4, ... hodnoty $g(0)$, $h(0)$, $g(1)$, $h(1)$, $g(2)$, $\ldots$, je primitivně rekurzívní. Dokažte. Platí analogické tvrzení i pro obecně rekurzívní funkce?

11. Dokažte, že obor hodnot funkce $x \mapsto x^2$ (tj. množina $\{0, 1, 4, 9, 16, \ldots\}$) je *PR*.

12. Přirozená čísla jsou nesoudělná, jestliže číslo 1 je jejich jediný společný dělitel. Dokažte, že relace nesoudělnosti je primitivně rekurzívní.

13. • Dokažte, že
    (a) Nekonečná množina přirozených čísel je rekurzívní, právě když je oborem hodnot rostoucí obecně rekurzívní funkce (jedné proměnné).
    (b) Obor hodnot libovolné obecně rekurzívní funkce je rekurzívně spočetná množina.

    Návod. V (a) $\Rightarrow$ sestrojte příslušnou funkci primitivní rekurzí podobně jako v důkazu tvrzení, že rostoucí posloupnost všech prvočísel je primitivně rekurzívní funkce. V $\Leftarrow$ nejprve dokažte, že je-li f rostoucí, pak vždy $f(x) \geq x$, a pak využijte omezené kvantifikátory. V (b) lze případně použít cvičení 5.

14. Nechť A je primitivně rekurzívní množina. Dokažte, že funkce $x \mapsto |A \cap x|$, která každému x přiřazuje počet prvků množiny A menších než x, je primitivně rekurzívní.

15. • Ke každé obecně rekurzívní funkci g s nekonečným oborem hodnot existuje prostá obecně rekurzívní funkce f se stejným oborem hodnot. Dokažte.

    Návod. Odvoďte f z g pomocí zobecněné (ordinální) rekurze.

16. Dokažte, že každý vývojový diagram je ekvivalentní s diagramem, který neobsahuje jiné instrukce než INC, CLR a porovnávání $X = Y?$.

17. Dokažte, že každý vývojový diagram je ekvivalentní s diagramem, který neobsahuje jiné instrukce než INC, DEC a $X = Y?$.

18. Dokažte, že každá množina instrukcí jazyka vývojových diagramů, ze které lze odvodit všechny ostatní instrukce, obsahuje jako podmnožinu buď množinu {INC, DEC, $X = 0?$} nebo jednu ze dvou množin jmenovaných ve cvičeních 16 a 17.

19. Dokažte, že v jazyce SAL (Simple Algorithmic Language, převzatém z Odifreddiho knihy a použitém v přednášce) lze bez užití cyklu **while** simulovat instrukce tvaru **if** *Cnd* **then** S, kde *Cnd* může mít kterýkoliv z tvarů $X \neq 0$, $X = 0$, $X \neq Y$, $X = Y$.

    Návod. Nejprve simulujte instrukce $U := \mathrm{sg}(X)$ a $U := |X - Y|$.

```
w := ⟨g_1(0)⟩
u := 0
for y do begin
    u := u + 1
    w := w * ⟨g_1(u)⟩
end
v := 0
for x do begin
    q := ⟨g_2(v)⟩
    u := 0
    for y do begin
        q := q * ⟨h((w)_{u+1}, (q)_u, v, u)⟩
        u := u + 1
    end
    v := v + 1
    w := q
end
z := (w)_y
```

Obrázek 1: Výpočet dvojné rekurze

20. • Nechť funkce f je definována z primitivně rekurzívních funkcí g_1, g_2 a h rovnostmi

$$\begin{aligned} f(0, y) &= g_1(y), \\ f(x+1, 0) &= g_2(x), \\ f(x+1, y+1) &= h(f(x, y+1), f(x+1, y), x, y). \end{aligned}$$

Popište neformální (tj. nezávislý na formalismu rekurzívních funkcí nebo určitého programovacího jazyka) algoritmus pro výpočet libovolné hodnoty funkce f. Pak dokažte, že f je primitivně rekurzívní.

Návod. Uvažujte program na obrázku 1. Program pracuje s posloupností w délky $y+1$. Ve stadiu v platí $w = \langle f(v, 0), .., f(v, y)\rangle$. Zdůvodněte, že tento program opravdu počítá funkci f a že může být považován za program v jazyce SAL.

21. • (Ackermannova funkce). Nechť posloupnost funkcí g_0, $g_1, \ldots$ je definována rovnostmi

$$\begin{aligned} g_0(y) &= y + 2, \\ g_{n+1}(0) &= g_n(1), \\ g_{n+1}(y+1) &= g_n(g_{n+1}(y)). \end{aligned}$$

(a) Popište neformální algoritmus pro výpočet hodnoty $g_n(y)$ a vypočtěte např. $g_3(1)$.
(b) Dokažte, že každá z funkcí g_n je primitivně rekurzívní.
(c) O funkci $[n, y] \mapsto g_n(y)$ dvou proměnných se dá dokázat (bohužel dost pracným důkazem), že *není* primitivně rekurzívní. Dokažte ale, že tato funkce má primitivně rekurzívní graf.
(d) Dokažte, že funkce z bodu (c) je obecně rekurzívní. Kdybychom první řádek definice změnili na $g_0(y) = y$, byla by dokonce primitivně rekurzívní.

Návod k (c). Napište program v jazyce SAL podobný programu ze cvičení 20. Nejdříve uvažte, že žádná funkční hodnota není 0 ani 1. Váš program nejprve přijme trojici $[x, y, z]$, o které má rozhodnout, zda $g_x(y) = z$. Dále pracuje s posloupností w délky $z - 1$. Ve stadiu v nechť posloupnost w je $\langle g_v(0), .., g_v(z-2) \rangle$ až na to, že hodnoty větší než z jsou nahrazeny nulami. Na začátku platí $w = \langle 2, 3, .., z \rangle$. Program opět může použít instrukce pro manipulace s posloupnostmi, a třeba také konstrukci **if–then–else**, nesmí ale použít **while**.