

Vypočítateľnosť

1. Funkcie a čiastočné funkcie, projekcie, skladanie funkcií, trojice číslovacích (párovacích) funkcií.

Def. Čiastočné (*totálne*) zobrazenie $z: A \rightarrow B$ je trojica (A, B, z) pre ktorú platí:

- $z \subseteq A \times B$
- Ku každému vstupu $a \in A$ existuje **najviac jeden** (*práve jeden*) výstup $b \in B$ taký, že $(a, b) \in z$.

Pozn. Ak je zobrazenie totálne, tak je aj čiastočné.

Označenie. A sú vstupy, B sú výstupy, f je graf.

Def. n-árna **čiastočná** funkcia na množine N je ľubovoľné čiastočné zobrazenie množiny N^k do množiny N. ($D(f) \subseteq N^k$)

Def. n-árna (**totálna**) funkcia na množine N je ľubovoľné (totálne) zobrazenie množiny N^k do množiny N. ($D(f) = N^k \approx$ všade definovaná funkcia)

Obor definície: $\text{Arg}(f) = \{x_1, \dots, x_k \mid \exists y : (x_1, \dots, x_k, y) \in f\}$

Obor hodnôt: $\text{Val}(f) = \{y \mid \exists x_1, \dots, x_k : (x_1, \dots, x_k, y) \in f\}$

Def. Rovnosť medzi dvomi výrazmi $A = B$ chápeme: bud' sú definovaní a majú rovnakú hodnotu alebo sú nedefinovaní.

Projekcia znižuje dimenziu, napr. pri premietnutí na os x, rušíme hodnoty na osi y – projekcia bodov $(2,3), (4,6), (2,1)$ na os x vyrobí body $(2), (4), (2)$.

Skladanie funkcií, zložená funkcia

Majme zobrazenia $f: A \rightarrow B$ a $g: C \rightarrow D$ také, že $H(f) \cap D(g) \neq \emptyset$. Množina $H(f) \cap D(g) = D(h)$ je definičným oborom *zloženej funkcie h*, ktorá vznikne *kompozíciou (skladaním)* funkcií f a g. Operáciu skladania funkcií označujeme o.

$$h(x) = (gof)(x) = g(f(x)), x \in f^{-1}(H(f) \cap D(g)).$$

Def. Nech f, g sú čiastočné funkcie na A. Budeme hovoriť, že **f je zúžením g** alebo **g je rozšírením f** ak $f \subseteq g$.

Ak g je rozšírením f, a g je totálna na A, potom budeme g nazývať **zúplnením** čiastočnej funkcie f na množine A.

Veta 3.43. Funkcie c , l , r tvoria trojicu číslovacích funkcií na množine \mathbb{N} , t.j. pre všetky $x, y \in \mathbb{N}$ platí

$$\begin{aligned} c(l(x), r(x)) &= x \\ l(c(x, y)) &= x \\ r(c(x, y)) &= y \end{aligned}$$

Vyjadrieme teraz funkcie c , l , r takým spôsobom, z ktorého bude zrejmá ich primitívna rekurzivnosť.

Hodnota $c(x, y)$ sa rovná počtu usporiadaných dvojíc, ktoré sú menšie ako dvojica (x, y) . Medzi týmito dvojicami je jedna dvojica so súčtom 0, dve dvojice so súčtom 1, ..., až $x + y$ dvojíc so súčtom $x + y - 1$, a ďalej x dvojíc so súčtom $x + y$, ale prvou zložkou menšou ako x . Preto platí

$$c(x, y) = 1 + 2 + \dots + (x + y) + x = \frac{(x + y)(x + y + 1)}{2} + x$$

Nech teraz $z = c(x, y)$. Postupne dostávame

$$\begin{aligned} 2z &= (x + y)(x + y + 1) + 2x \\ 8z + 1 &= (2x + 2y)(2x + 2y + 2) + 8x + 1 \\ 8z + 1 &= (2x + 2y + 1)^2 + 8x \\ 8z + 1 &= (2x + 2y + 3)^2 - 8x - 8 \end{aligned}$$

Odtiaľ dostávame nerovnosti

$$\begin{aligned} (2x + 2y + 1)^2 &\leq 8z + 1 < (2x + 2y + 3)^2 \\ 2x + 2y + 1 &\leq \lfloor \sqrt{8z + 1} \rfloor < 2x + 2y + 3 \\ x + y + 1 &\leq \left\lfloor \frac{\lfloor \sqrt{8z + 1} \rfloor + 1}{2} \right\rfloor < x + y + 2 \end{aligned}$$

Preto platí

$$x + y + 1 = \left\lfloor \frac{\lfloor \sqrt{8z + 1} \rfloor + 1}{2} \right\rfloor$$

Pretože $2x = 2z - (x + y)(x + y + 1)$, platí aj

$$\begin{aligned} l(z) &= x = z - \frac{(x + y)(x + y + 1)}{2} = z - \left\lfloor \frac{\left\lfloor \frac{\lfloor \sqrt{8z + 1} \rfloor + 1}{2} \right\rfloor \cdot \left\lfloor \frac{\lfloor \sqrt{8z + 1} \rfloor + 1}{2} \right\rfloor}{2} \right\rfloor \\ r(z) &= y = \left\lfloor \frac{\lfloor \sqrt{8z + 1} \rfloor + 1}{2} \right\rfloor - (l(z) + 1) \end{aligned}$$

Veta 3.44. Pre funkcie c , l , r platí

$$\begin{aligned} c(x, y) &= \frac{(x + y)(x + y + 1)}{2} + x \\ l(x) &= x - \left\lfloor \frac{\left\lfloor \frac{\lfloor \sqrt{8x + 1} \rfloor + 1}{2} \right\rfloor \cdot \left\lfloor \frac{\lfloor \sqrt{8x + 1} \rfloor + 1}{2} \right\rfloor}{2} \right\rfloor \\ r(x) &= \left\lfloor \frac{\lfloor \sqrt{8x + 1} \rfloor + 1}{2} \right\rfloor - (l(x) + 1) \end{aligned}$$

a teda tieto funkcie sú primitívne rekurzívne.

Teraz môžeme definovať číslovacie funkcie c^n , $c_{n,1}, \dots, c_{n,n}$ pre ľubovoľné $n \in \mathbb{N}$, $n \neq 0$.

Definícia 3.45. Pre každé $n \in \mathbb{N}$, $n \neq 0$, $i = 1, \dots, n$ a pre všetky x_1, \dots, x_n , $x \in \mathbb{N}$

$$\begin{aligned} c^1(x) &= I_1^1(x) \\ c_{1,1}(x) &= I_1^1(x) \end{aligned}$$

$$c^{n+1}(x_1, \dots, x_n, x) = c(c^n(x_1, \dots, x_n), x)$$

$$\begin{aligned} c_{n+1,i}(x) &= c_{n,i}(l(x)) \\ c_{n+1,n+1}(x) &= r(x) \end{aligned}$$

Veta 3.46. Funkcie z definície 3.45 sú primitívne rekurzívne.

2. Univerzálné funkcie a čiastočné funkcie. + univerzálné množiny

Definícia. Budeme hovoriť, že funkcia dvoch premenných $U : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ je univerzálnou funkciou pre triedu T funkcií jednej premennej ($f : \mathbb{N} \rightarrow \mathbb{N}$) ak

- Pre každé prirodzené číslo funkcia $U_n : x \rightarrow U(n, x)$ patrí do T .
- Pre každé $f \in T$ existuje n také, že pre všetky $x \in \mathbb{N}$ bude $f(x) = U_n(x) = U(n, x)$

Poznámka. Ak T má univerzálnu funkciu, tak $|T| \leq \aleph_0$.

Poznámka. Ak $f(x) = U(n, x)$, tak n sa volá číslo funkcie f .

Poznámka. Funkcia z T môže mať viacero (aj nekonečne veľa) čísel.

Veta 3.2.1 Existuje vypočítateľná funkcia dvoch premenných, ktorá je univerzálnou funkciou pre triedu vypočítateľných funkcií jednej premennej.

Dôkaz. O teste $z \in \Sigma^*$ vieme rozhodnúť, či je to výpočet, alebo nie. Potom z nazývame program. Navyše vieme algoritmicky rozhodnúť, či počíta funkciu jednej premennej. Teda máme generátor programov. Máme programy p_0, p_1, \dots . Iné funkcie vypočítateľné nie sú (nemáme na nich program).

$U(n, x)$ počítame nasledovne – vygenerujem program s číslom n , čiže p_n . Do neho dosadím číslo x . Ak dá výsledok, mám hodnotu, ináč je to nedefinované. U môže byť aj čiastočná funkcia.

Veta 3.3.1 Neexistuje totálna vypočítateľná funkcia dvoch premenných, ktorá je univerzálna pre triedu všetkých totálnych vypočítateľných funkcií jednej premennej.

Dôkaz. Diagonalizácia.

Zoberieme všetky totálne vypočítateľné, zoradíme (f_1, f_2, \dots) , spravíme totálnu vypočítateľnú, ktorá tam nie je $U(x, x) + 1$ – nie je rovná žiadnej f_i .

Veta 4.0.2 Existuje vypočítateľná funkcia $d : \mathbb{N} \rightarrow \mathbb{N}$, pre ktorú platí: Pre každú vypočítateľnú funkciu $f : \mathbb{N} \rightarrow \mathbb{N}$ existuje číslo n také, že platí $f(n) = d(n)$.

Dôkaz. $d(n) = U(n, n)$.

Veta 4.0.3 Existuje vypočítateľná funkcia $f : \mathbb{N} \rightarrow \mathbb{N}$, ktorá nemá vypočítateľné zúplnenie na \mathbb{N} .

Dôkaz. Nech $f'(n)$ je vypočítateľné zúplnenie funkcie $f(n)$, potom $\exists n : f'(n) = d(n) = U(n, n)$.

$$f'(n) = d(n).$$

$$U(n, n) + 1 \neq U(n, n), U(n, n)$$

je definovaná.
definovaná \neq nedefinovaná $U(n, n)$ nie je definovaná.

Veta 4.0.6 Existuje vypočítateľná funkcia $d_2 : \mathbb{N} \rightarrow \{0, 1\}$ ktorá nemá vypočítateľné zúplnenie na množine vstuov na \mathbb{N} .

Dôkaz. $d_2(x) = sg(U(x, x)) = 1$ ak $U(x, x) = 0$, ak $U(x, x) > 0$ je to 0 a ak je to nedefinované, tak je to nedefinované.

Definícia. Dve disjunktné množiny X, Y sú oddeliteľné množinou C ak $X \subseteq C$ a $C \cap Y = \emptyset$.

Veta 4.0.7 Existujú dve disjunktné rekurzívne očíslovateľné množiny, ktoré nie sú oddeliteľné žiadnou rekurzívnu množinou.

Dôkaz. Nech $X = \{x | d_2(x) = 1\}$ a $Y = \{x | d_2(x) = 0\}$. Obidve sú rekurzívne oči slovateľné. Nech $x \subseteq C$, $d_2(x) = 1$ nemení hodnotu, $C \cap T = \emptyset$ $d_2(x) = 0$ nemení hodnotu. Potom $\chi_C(s) = 1$ ak $x \in C$ a 0 ak $x \notin C$. Je to zúplnenie $d_w(x)$ a to nemôže byť vypočítateľné. Z toho vyplýva, že C nie je rekurzívne.

Definícia. Budeme hovoriť, že funkcia $n+1$ premenných $U : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ je univerzálnou funkciou pre triedu T funkcií n premenných, ak

- a) pre každé $k \in \mathbb{N}$ funkcia $U_k : x_1, \dots, x_n \rightarrow U(k, x_1, \dots, x_n)$ patrí do T .
- b) pre každú $f \in T$ sa nájde $k \in \mathbb{N}$ také, že pre všetky $x_1 \dots x_n \in \mathbb{N}^n$ bude $f(x_1, \dots, x_n) = U_k(x_1, \dots, x_n) = U(k, x_1, \dots, x_n)$. (k nazývame číslom funkcie f).

(Nasledujúce univerzálné funkcie sú pre registrované stroje, ktoré sú vysvetlené neskôr v 5 a 6.)

Definícia. Pre každé číslo $n \in \mathbb{N}$ definujeme $n+1$ -árnu funkciu predpisom $\forall z, x_1, \dots, x_n \in \mathbb{N}. m_{univ}^{(n+1)}(z, x_1, \dots, x_n) = \phi_z^n(x_1, \dots, x_n)$.

Veta 6.0.15 Pre každé priorodené číslo je čiastočná funkcia $m_{univ}^{(n+1)}(y, x_1, \dots, x_n)$ univerzálna funkcia pre množinu všetkých n -árnych M -vypočítateľných (na Minského stroji) funkcií.

Dôkaz. Každé $k \in \mathbb{N}$ je číslom nejakého stroja.

Veta 6.0.16 Pre každé prirodzené číslo $\forall z, x_1, \dots, x_n \in \mathbb{N}$ platí

$$m_{univ}^{(n+1)}(z, x_1, \dots, x_n) = m_{obs}(min(y, m_{vyp}^{(n+1)}(y, z, x_1, \dots, x_n) = 0))$$

Dôkaz. Je opäť triviálny.

Veta 6.0.17 Pre každé prirodzené $n \in \mathbb{N}$ je funkcia $m_{univ}^{(n+1)}$ je čiastočne rekurzívna univerzálna funkcia pre množinu $CRF^{(n)}$.

Dôkaz. a) $m_{univ}^{(n+1)} \in CRF$

b) Univerzálnosť pre $CRF^{(n)}$. $f(x_1, \dots, x_n) \in CRF^{(n)} \rightarrow f(x_1, \dots, x_n)$ je M vypočítateľná $\rightarrow \exists$ stroj z : $f(x_1, \dots, x_n) = \phi_z^n(x_1, \dots, x_n)$.

Stroj z má číslo a a teda $f(x_1, \dots, x_n) = m_{univ}^{(n+1)}(a, x_1, \dots, x_n) = m_{obs}(min(y, m_{vyp}^{(n+2)}(y, a, x_1, \dots, x_n) = 0))$;

Poznámka. $CRF^{(n)}$ = všetkým n -árnym funkciám vypočítateľným na minského stroja.

Definícia. Budeme hovoriť, že množina $W \subseteq \mathbb{N} \times \mathbb{N}$ je univerzálnou množinou pre triedu množín P prodmnožín prirodzených čísel, ak

- a) Pre každé $n \in N$ existuje množina $W_n = \{x | (n, x) \in W\} \in P$
- b) iných množín v P nict: $(\forall M \in P)(\exists n \in \mathbb{N})M = W_n$

Veta 3.2.2 Existuje rekurzívne očíslovateľná množina dvojíc prirodzených čísel, ktorá je univerzálna pre triedu všetkých rekurzívne očíslovateľných množín prirodzených čísel.

Dôkaz. M je rekurzívne očíslovateľná – má generátor. M je definičným oborom vypočítateľnej funkcie. Je tiež oborom hodnôt vypočítateľnej funkcie. Tiež $x \in M \Leftrightarrow (\exists y)(x, y) \in Q$

$$D(f_i) = M_i f_i$$

$(n, x) \in W \Leftrightarrow U(n, x)$ je definovaná.

Je generovateľná – diagonálne.

3. Primitívne rekurzívne, rekurzívne funkcie a čiastočne rekurzívne (vypočítateľné) funkcie (na množine N).

Definícia. Funkcia $f : N^k \rightarrow N$ sa nazýva vypočítateľnou, ak existuje algoritmus \mathcal{A} , ktorý ju vypočíta v nasledovnom zmysle:

- ak je hodnota $f(x_1, \dots, x_k)$ definovaná pre $x_i \in N$, potom sa algoritmus \mathcal{A} zastaví na vstupe x_1, \dots, x_k a vytlačí $f(x_1, \dots, x_k)$.
- Ak je hodnota $f(x_1, \dots, x_k)$ nedefinovaná pre x_1, \dots, x_k , potom sa algoritmus nezastaví na vstupe x_1, \dots, x_k .

Príklad. $Sg(x) = \text{if } x = 0 \text{ then } 0 \text{ else } 1$

$$\tilde{S}g(x) = \text{if } x = 0 \text{ then } 0 \text{ else } 1$$

$$\tilde{S}g(x) = 1 - Sg(x)$$

Poznámka. $f(x_1, \dots, x_k)$ nie je definovaná, práve vtedy keď algoritmus \mathcal{A} sa zacyklí alebo zastane a nič nedá na výstup.

Veta 3.1.1 Funkcia $f : N \rightarrow N$ je vypočítateľná vtedy a len vtedy, keď jej graf $f = \{(x, y) | f(x) \text{ je definované a } f(x) = y\} \subseteq N \times N$ je rekurzívne očíslovateľná množina.

Dôkaz.

- \Rightarrow Nech $f : N \rightarrow N$ je vypočítateľná (existuje algoritmus, čo počíta f). Diagonálne paralelne pustíme naraz všetky algoritmy, ktoré skončia, tie vypisujem.
 \Leftarrow Generujem a čakám na x , ak $x=a$ tak vypíšem y

Definícia. Nech $f : N \rightarrow N$ je čiastočná funkcia a nech $A \subseteq N, B \subseteq N$. Potom $f(A) = \{y | (\exists x) x \in A \wedge (x, y) \in f\}$ (obraz). $f^{-1}(B) = \{x | (\exists y) y \in B \wedge (x, y) \in f\}$ (vzor).

Veta 3.1.2 Nech $f : N \rightarrow N$ je vypočítateľná funkcia. A nech $A \subseteq N, B \subseteq N$ sú rekurzívne očíslovateľné množiny. Potom aj obraz $f(A)$ a vzor $f^{-1}(B)$ sú rekurzívne očíslovateľné.

Dôkaz. f je vypočítateľná – vieme jej graf generovať. Zoberieme generátor A a generátor grafu. Naraz pustíme oba generátory a zapamätvame si výsledky, a keď nájdeme match, tak to hodíme na výstup. Podobne to funguje aj s generátorom B ...
 $(x \alpha (x,y) \text{ kde } x=x')$

Poznámka. $(\text{graf } f) \cap (N \times B) \xrightarrow{\text{projekcia}} = f^{-1}(B)$ $(\text{graf } f) \cap (A \times N) \xrightarrow{\text{projekcia}} = f(A)$

Primitívne rekurzívne funkcie \subset všeobecné rekurzívne funkcie \subset čiastočne rekurzívne funkcie.

Máme k dispozícii:

- $0 = \lambda(0)$
- $s(x) = \lambda x(x + 1)$ (succesor)
- $I_i^n(x_1, \dots, x_i, \dots, x_n) = x + i - \lambda x_1, \dots, x_n(x_i)$.
- primitívna rekurzia (for cyklus – parameter, koľko krát sa cyklus opakuje)
- regulárna substitúcia – strom – potiaľto sú primitívne rekurzívne.
- minimalizácia pre y ($f(y, x_1, \dots, x_n) = 0$) – lineárne prehľadávanie – odiaľto sú čiastočne rekurzívne

Definícia. (Regulárna substitúcia/skladanie)

Nech f je n -árna (čiastočná) funkcia ($n > 0$). Nech g, f_1, \dots, f_n sú už m -árne (čiastočné) funkcie. Budeme hovoriť, že (čiastočná) funkcia g vzniká skladaním (substitúciou) z (čiastočných) funkcií f, f_1, \dots, f_n a písat $g = \mathbb{S}(f, f_1, \dots, f_n)$ ak pre všetky x_1, \dots, x_n platí, že $g(x_1, \dots, x_n) = f(f_1(x_1, \dots, x_m), \dots, f_n(x_1, \dots, x_m))$.

Definícia. (Primitívna rekurzia)

Nech f je $(n+1)$ -árna a g je $(n+2)$ -árna a h je n -árna čiastočná funkcia na \mathbb{N} , kde $n \geq 0$. Hovoríme, že f vzniká z g , h operáciou primitívnej rekurzie, a píšeme že $f = R(g, h)$ ak pre všetky $x_1, \dots, x_n, y \in \mathbb{N}$ platí:

1. $f(0, x_1, \dots, x_n) = h(x_1, \dots, x_n)$
2. $f(y+1, x_1, \dots, x_n) = g(y, f(y, x_1, \dots, x_n), x_1, \dots, x_n)$

Príklad. $f(y, x) = x^y$.

$$f(0, x) = x^0 = 1 = \lambda x(1).$$

$$f(y+1, x) = x^{y+1} = x^y \cdot x = g(y, x^y, x) = \lambda \alpha, \beta, \gamma (\beta \cdot \gamma).$$

$$\lambda x, y (x^y) = R(\lambda \alpha, \beta, \gamma (\beta \cdot \gamma), \lambda x(1)).$$

Cvičenie. Ukážte, že ak g, h sú totálne, tak aj $R(g, h)$ je totálna funkcia.

Riešenie. Indukciu cez y . Prvý krok je očividný, lebo h je definovaná. Druhý krok – pre y je to definované, mám definované vstupy a g je totálne, tak mám aj definovaný výstup, takže je to totálne.

Definícia. Budeme hovoriť, že funkcia f je primitívne rekurzívna, ak vzniká z funkcií $0, s, I_m^n$ konečným počtom operácií skladania \mathcal{S} a primitívnej rekurzie \mathcal{R} , t.j ak existuje konečná postupnosť funkcií f_1, \dots, f_k taká, že $f = f_k$ a $\forall i \in \{1, \dots, k\}$ platí jedna z nasledovných možností:

- a) $f_i = 0$
- b) $f_i = s$
- c) $f_i = I_m^n$
- d) $f_i = \mathcal{S}^{m+1}(f_j, f_{j_1}, \dots, f_{j_m}) \quad 1 \leq j, j_1, \dots, j_m < i$
- e) $f_i = \mathcal{R}(f_a, f_b), a, b < i$

Veta 5.0.1 Množina PRF primitívne rekurzívnych funkcií je najmenšia množina (\subseteq) obsahujúca funkcie $0, s, I_m^n$ a uzavretá na operácie \mathcal{S}, \mathcal{R} .

Dôkaz. Sporom. Nech M obsahuje $0, s, I$ a je uzavretá na \mathcal{S}, \mathcal{R} . Nech $f \in PRF$, potom $f \in M$? Áno, lebo ak zoberieme postupnosť f_1, \dots, f_k a tá je dobrá aj v M . QED

Veta 5.0.2 Funkcia, ktorá vznikla konečným počtom operácií skladania a primitívnej rekurzie z primitívne rekurzívnych funkcií je primitívne rekurzívna.

Veta 5.0.3 Nech množina funkcií M obsahuje všetky I_m^n a je uzavretá na regulárnu substitúciu (skladanie). Nech funkcia $f(x_1, \dots, x_m), m \neq 0$ patrí do M . Nech K_1, \dots, K_m je libovoľná postupnosť prvkov množiny $\{1, \dots, n\}$. Potom aj funkcia $g(x_1, \dots, x_n) = f(x_{K_1}, \dots, x_{K_m}) \in M$.

Dôkaz. $f = S^{m+1}(f, I_{K_1}^n, \dots, I_{K_m}^n)$.

Veta 5.0.5 Nech množina funkcií M obsahuje všetky I_m^n , všetky unárne konštanty K_a^1 a je uzavretá na regulárnu substitúciu. Nech funkcia $f(x_1, \dots, x_n), n \neq 0$ patrí do M . Nech $a_1, \dots, a_k \in \mathbb{N}, k < n$. Potom aj funkcia $g(x_k + 1, \dots, x_n) = f(a_1, \dots, a_k, x_{k+1}, \dots, x_n)$ patrí do M .

Dôkaz. $S^{n+1}(f, \mathcal{S}^2(K_{a_1}, I_1^{n-k}), \dots, \mathcal{S}^2(K_{a_k}, I_k^{n-k}), I_1^{n-k}, \dots, I_{n-k}^{n-k})$

Poznámka. $k < n$, lebo argument funkcie musí byť aspoň jedna premenná. Čo ak chceme zapchať všetkých? Ako to bude vyzerat? Potom $f(a_1, \dots, a_n) = K_{f(a_1, \dots, a_n)}^0$.

Veta 3.8. Všetky konštantné funkcie sú primitívne rekurzívne.

Dôkaz: Nulárna funkcia 0 je primitívne rekurzívna podľa definície. Unárna funkcia $o = \lambda x(0)$ vzniká primitívou rekurziou z funkcií 0, I_2^2 , lebo pre všetky $x \in \mathbb{N}$ platí

$$\begin{aligned} o(0) &= 0 \\ o(x+1) &= I_2^2(x, o(x)) \end{aligned}$$

Funkcia $\lambda x(k+1)$ vzniká skladaním funkcií $s, \lambda x(k)$:

$$\lambda x(k+1) = \mathcal{S}^2(s, \lambda x(k))$$

Tým sme indukciou dokázali, že všetky unárne konštantné funkcie sú primitívne rekurzívne. Podľa cvičenia 2.11 sú potom aj všetky n -árne ($n \neq 0$) konštantné funkcie primitívne rekurzívne a podľa vety 2.14 sú aj všetky konštanty primitívne rekurzívne. \square

Veta 3.10. Funkcie $\lambda xy(x+y), \lambda xy(x.y), \lambda xy(y^x)$ sú primitívne rekurzívne.

Dôkaz: Pre všetky $x, y \in \mathbb{N}$ platí

$$\begin{aligned} 0+y &= I_1^1(y) \\ (x+1)+y &= s(I_2^3(x, x+y, y)) \end{aligned}$$

a teda funkcia $x+y$ vzniká z primitívne rekurzívnych funkcií I_1^1 a $\mathcal{S}^2(s, I_2^3)$ operáciou primitívnej rekurzie, teda je sama primitívne rekurzívna. Z už dokázaného a zo vzťahov

$$\begin{aligned} 0.y &= o(y) \\ (x+1).y &= I_2^3(x, x.y, y) + I_3^3(x, x.y, y) \end{aligned}$$

vyplýva primitívna rekurzívnosť funkcie $x.y$ a z nej a zo vzťahov

$$\begin{aligned} y^0 &= K_1^1(y) \\ y^{x+1} &= I_2^3(x, y^x, y) \cdot I_3^3(x, y^x, y) \end{aligned}$$

(kde sme použili označenie K_1^1 pre unárnu funkciu identicky rovnajúcu sa jednej) vyplýva primitívna rekurzívnosť funkcie y^x . \square

Poznámka 3.11. Funkcia $\lambda xy(x^y)$ je tiež primitívne rekurzívna, lebo platí

$$\lambda xy(x^y) = \mathcal{S}^3(\lambda xy(y^x), I_2^2, I_1^2)$$

Veta 5.0.11 (Veta o IFnutí)

Nech $f_1, \dots, f_{s+1}, h_1, \dots, h_s \in PRF^{(n)}$. Nech pre žiadne $x_1, \dots, x_n \in \mathbb{N}^n$ sa žiadne dve z funkcií h_1, \dots, h_s nerovnajú súčasne nule. Potom funkcia f definovaná predpisom $f(x_1, \dots, x_n) = f_1(x_1, \dots, x_n)$ ak $h_1(x_1, \dots, x_n) > 0$
 $f(x_1, \dots, x_n) = f_2(x_1, \dots, x_n)$ ak $h_2(x_1, \dots, x_n) > 0$
 \vdots
 $f(x_1, \dots, x_n) = f_s(x_1, \dots, x_n)$ ak $h_s(x_1, \dots, x_n) > 0$
 $f(x_1, \dots, x_n) = f_{s+1}(x_1, \dots, x_n)$ ináč
je PRF .

Dôkaz. $f(\bar{x}) = f_1(\bar{x})\overline{sg}(h_1(\bar{x})) + \dots + f_s(\bar{x})(h_s(\bar{x})) + f_{s+1}(\bar{x}).sg(h_1, h_2, \dots)$

Lema 3.17. Nech $g(y, x_1, \dots, x_n)$ je primitívne rekurzívna funkcia. Potom aj funkcie f'_1 a f'_2 dané predpismi

$$\begin{aligned} f'_1(y, z, x_1, \dots, x_n) &= \sum_{i=z}^{y+z} g(i, x_1, \dots, x_n) \\ f'_2(y, z, x_1, \dots, x_n) &= \prod_{i=z}^{y+z} g(i, x_1, \dots, x_n) \end{aligned}$$

sú primitívne rekurzívne.

Dôkaz: Platí

$$\begin{aligned} f'_1(0, z, x_1, \dots, x_n) &= g(z, x_1, \dots, x_n), \\ f'_1(y+1, z, x_1, \dots, x_n) &= f'_1(y, z, x_1, \dots, x_n) + g(y+z+1, x_1, \dots, x_n). \end{aligned}$$

Teda funkcia f'_1 vzniká primitívou rekurziou z primitívne rekurzívnych funkcií, čiže je aj sama primitívne rekurzívna. Dôkaz pre funkciu f'_2 je úplne analogický. \square

Veta 3.18. Nech sú $g(y, x_1, \dots, x_n)$, $h(x_1, \dots, x_n)$, $k(x_1, \dots, x_n)$ primitívne rekurzívne funkcie. Potom aj funkcie f_1 , f_2 dané predpismi

$$f_1(x_1, \dots, x_n) = \begin{cases} \sum_{i=h(x_1, \dots, x_n)}^{k(x_1, \dots, x_n)} g(i, x_1, \dots, x_n), & \text{ak } h(x_1, \dots, x_n) \leq k(x_1, \dots, x_n) \\ 0, & \text{ak } h(x_1, \dots, x_n) > k(x_1, \dots, x_n) \end{cases}$$

$$f_2(x_1, \dots, x_n) = \begin{cases} \prod_{i=h(x_1, \dots, x_n)}^{k(x_1, \dots, x_n)} g(i, x_1, \dots, x_n), & \text{ak } h(x_1, \dots, x_n) \leq k(x_1, \dots, x_n) \\ 1, & \text{ak } h(x_1, \dots, x_n) > k(x_1, \dots, x_n) \end{cases}$$

sú primitívne rekurzívne.

Dôkaz: Nech f'_1 , f'_2 sú funkcie z lemy 3.17. Označme ďalej h' , k' funkcie dané predpismi

$$h'(x_1, \dots, x_n) = h(x_1, \dots, x_n) \dot{-} k(x_1, \dots, x_n)$$

$$k'(x_1, \dots, x_n) = k(x_1, \dots, x_n) \dot{-} h(x_1, \dots, x_n)$$

Pre všetky $x_1, \dots, x_n \in \mathbb{N}$ platí

$$f_1(x_1, \dots, x_n) = f'_1(k'(x_1, \dots, x_n), h(x_1, \dots, x_n), x_1, \dots, x_n) \cdot \overline{\text{sg}}(h'(x_1, \dots, x_n))$$

$$f_2(x_1, \dots, x_n) = f'_2(k'(x_1, \dots, x_n), h(x_1, \dots, x_n), x_1, \dots, x_n) \cdot \overline{\text{sg}}(h'(x_1, \dots, x_n)) +$$

$$+ \text{sg}(h'(x_1, \dots, x_n))$$

Tým sme vyjadrili funkcie f_1 , f_2 pomocou operácie skladania funkcií a primitívne rekurzívnych funkcií, teda sme dokázali aj ich primitívnu rekurzívnosť. \square

Veta 3.28. Funkcia

$$\lfloor x/y \rfloor = \begin{cases} 0, & \text{ak } y = 0, \\ (\text{dolná}) \text{ celá časť podielu } \frac{x}{y}, & \text{ak } y \neq 0 \end{cases}$$

je primitívne rekurzívna.

Dôkaz: Stačí uvážiť, že pre všetky $x, y \in \mathbb{N}$ platí

$$\lfloor x/y \rfloor = \text{sg}(y) \cdot \sum_{i=1}^x \overline{\text{sg}}(i \cdot y \dot{-} x) \quad \square$$

Pre funkciu budeme používať aj označenia $\left\lfloor \frac{x}{y} \right\rfloor$, $x \text{ DIV } y$.

Definícia 3.22. Nech f je n -árna čiastočná funkcia, g je $(n+1)$ -árna čiastočná funkcia. Budeme hovoriť, že čiastočná funkcia f vzniká z čiastočnej funkcie g *minimalizáciou* a písť $f = \mathcal{M}(g)$ alebo

$$f(x_1, \dots, x_n) = \mu_y(g(y, x_1, \dots, x_n) = 0), \quad (3.22.1)$$

ak pre všetky $x_1, \dots, x_n \in \mathbb{N}$ platí $f(x_1, \dots, x_n) = y$ práve vtedy, keď pre všetky $z < y$ je $g(z, x_1, \dots, x_n)$ definované a kladné a súčasne $g(y, x_1, \dots, x_n) = 0$.

Definícia 3.23. Ak $f = \mathcal{M}(g)$ a f , g sú totálne funkcie, budeme hovoriť, že funkcia f vzniká z funkcie g *regulárной minimalizáciou*.

Poznámka. Minimalizácia dáva výsledok (končí) ak existuje y . Operácia minimalizácie a regulárnej minimalizácie nezachováva primitívnu rekurzívnosť

Poznámka. Ak funkcia g je totálna, potom $f(\bar{x}) \downarrow \leftrightarrow (\exists y)(g(y, \bar{x}) = 0)$

Veta 3.26. Nech $g(y, x_1, \dots, x_n)$, $h(x_1, \dots, x_n)$ sú primitívne rekurzívne funkcie a nech pre každé $x_1, \dots, x_n \in \mathbb{N}$ existuje také $z \leq h(x_1, \dots, x_n)$, že $g(z, x_1, \dots, x_n) = 0$. Potom aj funkcia $f(x_1, \dots, x_n) = \mu_y(g(y, x_1, \dots, x_n) = 0)$ je primitívne rekurzívna.

Dôkaz: Nech

$$f_1(z, x_1, \dots, x_n) = \prod_{i=0}^z g(i, x_1, \dots, x_n)$$

V postupnosti $f_1(0, x_1, \dots, x_n)$, $f_1(1, x_1, \dots, x_n)$, $f_1(2, x_1, \dots, x_n)$, \dots je presne

$$\mu_y(g(y, x_1, \dots, x_n) = 0)$$

nenulových členov, a všetky členy počnúc $f_1(h(x_1, \dots, x_n), x_1, \dots, x_n)$ sú nulové. Preto platí

$$f(x_1, \dots, x_n) = \sum_{i=0}^{h(x_1, \dots, x_n)} \text{sg}(f_1(i, x_1, \dots, x_n)) \quad \square$$

Poznámka. Veta 3.26 sa nazýva aj veta o ohraničenej minimalizácii.

Veta 3.37. Funkcie

$$\begin{aligned} \lfloor \sqrt{x} \rfloor &= \mu_z (\overline{\text{sg}}((z+1)^2 - x) = 0) \\ q(x) &= x - \lfloor \sqrt{x} \rfloor^2 \end{aligned}$$

sú primitívne rekurzívne.

Dôkaz: Veta bezprostredne vyplýva zo vzťahov v nej uvedených a z nerovnosti $\lfloor \sqrt{x} \rfloor \leq x$. \square

Veta 3.35. Funkcia

$$\text{ex}(x, y) = \begin{cases} \text{exponent prvočísla } p(x) \text{ v rozklade čísla } y \text{ na prvočinitele,} & \text{ak } y \neq 0 \\ 0, & \text{ak } y = 0 \end{cases}$$

je primitívne rekurzívna.

Dôkaz: Platí $\text{ex}(x, y) \leq y$ a $\text{ex}(x, y) = \mu_u(\text{sg}(y). \text{div}(y, (p(x))^{u+1}) = 0)$. Teraz stačí použiť veta 3.26. \square

Def. Funkciu $f: \mathbb{N}^n \rightarrow \mathbb{N}$ budeme nazývať **všeobecne rekurzívnu**, ak vzniká z funkcií **0**, **s(x) = $\lambda x(x + 1)$** , **I_m^n ($1 \leq m \leq n$)**, konečným počtom operácií **skladania S** (reg. substitúcia), **primitívnej rekurzie R** a regulárnej **minimalizácii**.

Def. Funkciu $f: \mathbb{N}^n \rightarrow \mathbb{N}$ budeme nazývať **čiastočne rekurzívnu**, ak vzniká z funkcií **0**, **s(x) = $\lambda x(x + 1)$** , **I_m^n ($1 \leq m \leq n$)**, konečným počtom operácií **skladania S** (reg. substitúcia), **primitívnej rekurzie R** a **minimalizácii**.

Veta. **PRF \subseteq VRF \subseteq ČRF**

Poznámka. Množiny nie sú vlastnými podmnožinami

Poznámka. Veta o ifnutí, \sum , \prod dokázaná pre ČRF platí aj pre PRF

Veta 6.0.18 $VRF = \text{totálne CRF}$

Dôkaz. $VRF - 0, +1, I, S, R$, regulárnu minimalizáciou (zachováva totálnosť)

Totálne CRF sú tie čiastočné, ktoré sú totálne.

$VRF \subseteq \text{Totálne CRF}$

$\text{Totálne CRF} \subseteq VRF$

Nech $f \in \text{totálne CRF}$. $F(x_1, \dots, x_n) = m_{obs}(\min(y, m_{vyp}^{(n+2)}(y, a, x_1, \dots, x_n) = 0))$. – máme tam PRF a aj totálnu, takže minimalizácia je regulárna.

Poznámka. V definícii by sa dalo požadovať, že môže byť iba jedna minimalizácia.

Veta 6.0.19 (Kleene, o normálnej forme) Nech $n \in \mathbb{N}$. Potom existujú také primitívne rekurzívne funkcie $u \in PRF^{(1)}$ a $v \in PRF^{(n+2)}$, že každú n -árnu čiastočne rekurzívnu funkciu $f \in PRF^{(n)}$ môžeme zapísť v tvare $f(x_1, \dots, x_n) = u(\min(y, v(y, k, x_1, \dots, x_n) = 0))$ pre nejaké $k \in \mathbb{N}$.

Dôkaz. $u = m_{obs}$, $v = m_{vyp}$.

4. Primitívne rekurzívne, rekurzívne a rekurzívne spočítateľné množiny a predikáty a ich vlastnosti.

Definícia 1.24. (a) Čiastočnou charakteristickou funkciou n -árnej relácie M na množine X nazveme takú n -árnu čiastočnú funkciu f na množine X , že pre všetky $x_1, \dots, x_n \in X$ platí:

$$f(x_1, \dots, x_n) = \begin{cases} 0, & \text{ak } (x_1, \dots, x_n) \in M \\ \uparrow, & \text{ak } (x_1, \dots, x_n) \notin M \end{cases}$$

(b) Charakteristickou funkciou n -árnej relácie M na množine X nazveme takú n -árnu funkciu f na množine X , že pre všetky $x_1, \dots, x_n \in X$ platí:

$$f(x_1, \dots, x_n) = \begin{cases} 0, & \text{ak } (x_1, \dots, x_n) \in M \\ 1, & \text{ak } (x_1, \dots, x_n) \notin M \end{cases}$$

Definícia. Množina $M \subseteq \mathbb{N}^k$ je rekurzívne vypočítateľná, ak jej charakteristická funkcia $\chi_M(x_1, \dots, x_k)$ je vypočítateľná: $\chi_M(x_1, \dots, x_k) = 1$ ak $x_1, \dots, x_k \in M$, a 0 ináč.

Definícia. Množina $M \subseteq \mathbb{N}^K$ je rekurzívne očíslovateľná (alebo rekurzívne spočítateľná – recursive enumerable). Ak existuje generujúci algoritmus, ktorý postupne tlačí všetky prvky množiny M a žiadne iné.

Poznámka. Generujúci algoritmus – vlastnosti:

- Nemá vstup, má len začiatok, nemá koniec práce.
- Každý pravok sa vytlačí za konečný čas.
- čas medzi dvoma výpismi je konečný, ale nemusí byť konštantný.
- Poradie vypísaných pravok nie je rozhodujúce.
- Tlač toho istého pravku sa môže zopakovať.
- Od istého momentu sa už nič nemusí vypísať.

Veta 2.1.3 Nech $M \subseteq \mathbb{N}$. Potom nasledovné tvrdenia sú ekvivalentné.

- M je rekurzívne očíslovateľné.
- M je definičným oborom (nejakej) vypočítateľnej funkcie. t.j. $(\exists f)(M = \text{Arg}(f))$
- M je oblasťou hodnôt vypočítateľnej funkcie. t.j. $(\exists f)(M = \text{Val}(f))$
- χ_M je vypočítateľná funkcia.

Dôkaz. • Nech M je rekurzívne vyčísliteľná. Existuje generujúci algoritmus, ktorý generuje $a_0, a_1, \dots, a_n, \dots$. Ak $f(x)$ je definovaný, x je generované, ak $f(x)$ nie je definovaný, x nie je generované. Nech $f(x) = \text{def} 1$ ak x je generované a ináč je to nedefinované. – je to vypočítateľná funkcia (čakám na x). Ak bude generovaný, vypočítam, ak nie – cyklus – nedefinovaná). Zároveň $f(x)$ je χ_M . Dokázali sme, že z a) vyplýva b) aj d).

- Nech $g(x) = \text{def} x$ ak x je generované, alebo je to nedefinované, ak x nie je generované. Je to vypočítateľná funkcia a dokázali sme že z a) vyplýva c).
- Nech f je vypočítateľná funkcia algoritmom B . Spustíme algoritmus pre všetky prirodzené čísla.

$B(0)$	$B(1)$	$B(2)$	\dots	$B(n)$
k_{00}	k_{10}	k_{20}		
k_{01}	k_{11}			
k_{02}				
\dots				
				k_{np}

-> posledný pravok pre výpočet $B(n)$

Pôjdeme po diagonálach (také, aby sme prešli každý krok) – generujeme vstup $M = \text{Arg}(f)$ čiže z b) aj d) vyplýva a). Tiež vieme generovať výstupnú hodnotu, čím sme dostali že $M = \text{Val}(f)$ a tým sme splnili, že z c) vyplýva a).

Poznámka. V prípade, že $M \subseteq \mathbb{N}^K$, potom c) musíme vyhodiť lebo k-tice nemôžu byť na výstupe a), b), d) sú ekvivalentné.

Veta 2.1.4 Nech $M \subseteq \mathbb{N}$. Množina M je rekurzívne očíslovateľná $\Leftrightarrow M = \emptyset$ alebo M je oblasťou hodnôt totálne vypočítateľnej funkcie.

Dôkaz. Nech M je rekurzívne očíslovateľná a neprázdna.

\Leftarrow Necháme bežať generátor. V každom takte "generátora" zvýšime n , a $f_M(n) = x$, kde x je posledné vypísané číslo z generátora. – je to vypočítateľná funkcia.

Aspoň jeden prvok vypadne, pretože množina M je neprázdna.

Na každý krok algoritmu buď vypadne prvok patriaci do f alebo nevypadne a f priradíme napr. posledne tlačený prvok

- \Rightarrow Nech máme funkciu, ktorá je totálna a vypočítateľná. Generujeme – najskôr generujeme $f(0)$, potom $f(1)$, potom $f(2), \dots$

Algoritmus vygeneruje všetky hodnoty množiny M

```
i:=0  
while true do  
    write(f(i))  
    i:=i+1
```

Veta 2.1.5 Zjednotenie, prienik rekurzívne očíslovateľných množín je rekurzívne očíslovateľná.

Dôkaz.

1. Zjednotenie: Chvíľku generujeme v jednom, chvíľku generuje v druhom a čo vypadne z ktoréhokolvek, to vypíšem.
2. Prienik: Chvíľku generujeme v jednom, chvíľku v druhom, to, čo vyplujú si zapamätáme, a ak raz sa stane, že boli oba vyplúté, tak to vyplújem. Skrátka treba pamäť.

Poznámka. Pre rozdiel to už neplatí.

Veta 2.1.6 $L \subset \mathbb{N}, K \subset \mathbb{N}$ sú rekurzívne očíslovateľné, potom $L \times K \subset \mathbb{N} \times \mathbb{N}$ je rekurzívne očíslovateľná.

Dôkaz. Generujem paralelne, pamätam si, vždy keď príde nový, vypíšem dvojice, ktoré s ním viem vytvoriť.

Veta 2.1.7

- a) Každá rekurzívna množina je rekurzívne očíslovateľná.
- b) Postova veta: Ak množina $M \subseteq \mathbb{N}$ ako aj jej doplnok je rekurzívne očíslovateľný, potom M je rekurzívna.

Dôkaz.

a)

```
x:=0  
while true do  
    if χ_M(x) then  
        write(x)  
    x:=x+1;
```

M je rekurzívna a keďže máme program, tak je aj rekurzívne očíslovateľná.

- b) Paralelne skúšame, či tam patrí tam alebo do doplnku, časom dostaneme výsledok z jedného a teda vieme výsledok vždy.

Definícia 4.1. (a) Nech $n \in \mathbb{N}$ je ľubovoľné. Množinu M n -tíc prirodzených čísel budeme nazývať (primitívne) rekurzívnu, ak jej charakteristická funkcia je (primitívne) rekurzívna.

(b) Množinu M prirodzených čísel budeme nazývať (primitívne) rekurzívnu, ak je množina M^1 (primitívne) rekurzívna.

Definícia 4.2. Nech $n \in \mathbb{N}$ je ľubovoľné. n -árny predikát $P(x_1, \dots, x_n)$ nazveme (primitívne) rekurzívny, ak jeho charakteristická funkcia je (primitívne) rekurzívna.

Veta 4.0.4 Existuje rekurzívne očíslovateľná množina, ktorá nie je rekurzívna.

Dôkaz. Nech $f(x)$ je vypočítateľná funkcia, ktorá nemá vypočítateľné zúplnenie. F - definičný obor f ($\text{Arg}(f) \neq \mathbb{N}$) F je rekurzívne očíslovateľná množina (je to definičný obor vypočítateľnej funkcie). F nie je rekurzívna množina.

Nech F je rekurzívna. Ak $x \in F$, tak vyhodí $f(x)$, ináč vyhodí 0, čo je vypočítateľné zúplnenie f , čo je spor.

Veta 3.0.9 Množina $M \subseteq \mathbb{N}$ je rekurzívne spočítateľná $\Leftrightarrow M$ je projekciou nejakej rekurzívnej množiny $Q \subseteq \mathbb{N} \times \mathbb{N}$, t.j. $x \in M \Leftrightarrow (\exists y)((x, y) \in Q)$

y je svedok, že x patrí M

Dôkaz.

\Leftarrow Nech Q je rekurzívna \Rightarrow je rekurzívne očíslovateľná \Rightarrow (lebo projekcia rek. očíslovateľnej množiny je rekurzívne očíslovateľná – vynechaním nepotrebných súradníč) M je rekurzívne spočítateľná.

\Rightarrow Nech M je rekurzívne spočítateľná. To znamená, že existuje generátor, ktorý M generuje. Spravíme dvojice (x, k) , $x \in M$ k je počet krokov za ktoré sa x vygeneruje. To je rekurzívne očíslovateľná množina Q .

Veta 2.1.1 Každá konečná množina $M \subseteq \mathbb{N}^k$ je rekurzívna (algoritmicke). Nech $M_1, M_2 \subseteq \mathbb{N}^k$ a M_1, M_2 sú rekurzívne. Potom $M_1 \cap M_2, M_1 \cup M_2$ a $M_1 \setminus M_2$ sú rekurzívne.

Dôkaz. a.) if $x = x_1$ alebo \dots alebo $x = x_1$ then 1 else 0 (Lineárne prehľadávanie - na konečný počet krokov dostanem odpoved)

b.) $\chi_{M_1}(x)$

$\chi_{M_2}(x)$

$\chi_{M_1 \cup M_2}(x) = \chi_{M_1}(x) \cdot \chi_{M_2}(x)$

$\chi_{M_1 \cap M_2}(x) = sg(\chi_{M_1}(x) + \chi_{M_2}(x))$

$\chi_{M_1 \setminus M_2}(x) = \chi_{M_1}(x) \cdot (1 - \chi_{M_2}(x))$

Veta 2.1.2 Nekonečná množina $M \subseteq \mathbb{N}$ je rekurzívna práve vtedy, keď je množinou hodnôt totálnej rastúcej vypočítateľnej funkcie.

Dôkaz. $\bullet \Rightarrow$. Nech nekonečná $M \subseteq \mathbb{N}$ je rekurzívna. $\chi_M(x) = 1$ ak $x \in M$ a 0 ináč. Treba vytvoriť $f(x)$. $x_0 < x_1 < x_2 < \dots < x_n < \dots$ nekonečná rastúca postupnosť prvkov z M .

$f(n) = x_n$

$f(n)$ má program.

```

1) p:=0
2) x:=0
3) if  $\chi_M(x)=1$  then p:=p+1
4) while p<n+1 do
5)     x:=x+1;
6)     if  $\chi_M(x)=1$  then p=p+1

```

Invariant riadok 3: x je posledné preverené číslo, p je počet kladných odpovedí medzi doteraz preverenými číslami. Na konci: $p = n + 1$ a invariant: $x = f(n + 1)$.

$\bullet \Leftarrow$ Máme totálnu rastúcu vypočítateľnú funkciu $f : \mathbb{N} \rightarrow \mathbb{N}$. Vieme, že $Val(f) = M$.

$x \in M$ ak $\exists n : x = f(n)$

$x \notin M$ ak $\exists n : f(n) < x < f(n + 1)$

O každom číseľ viem či patrí alebo nepatrí. Bud nájdem n alebo padnem do intervalu medzi dve hodnoty a viem, že som skončil.

Veta 4.7. Nech $n \in \mathbb{N}$ je ľubovoľné a nech $M_1 \subseteq \mathbb{N}^n, M_2 \subseteq \mathbb{N}^n$ sú (primitívne) rekurzívne množiny. Potom sú aj množiny $M_1 \cup M_2, M_1 \cap M_2, M_1 \setminus M_2, \mathbb{N}^n \setminus M_1$ (primitívne) rekurzívne.

Dôkaz: Ak sú $\chi_1(x_1, \dots, x_n)$, resp. $\chi_2(x_1, \dots, x_n)$ charakteristické funkcie množín M_1 , resp. M_2 , tak $1 - \chi_1(x_1, \dots, x_n), \chi_1(x_1, \dots, x_n) \cdot \chi_2(x_1, \dots, x_n)$ sú charakteristické funkcie množín $\mathbb{N}^n \setminus M_1, M_1 \cup M_2$ a sú zrejme (primitívne) rekurzívne. Pretože rozdiel a prienik množín možno vyjadriť pomocou operácií zjednotenia a komplementu, sú aj množiny $M_1 \cap M_2, M_1 \setminus M_2$ (primitívne) rekurzívne. \square

Definícia 1.37. Nech $P(x_1, \dots, x_n)$ je n -árny predikát na množine X .

- (a) Čiastočnou charakteristickou funkciou predikátu $P(x_1, \dots, x_n)$ je taká n -árna čiastočná funkcia f na množine X , že pre všetky $a_1, \dots, a_n \in X$ platí:

$$f(a_1, \dots, a_n) = \begin{cases} 0, & \text{ak } P(a_1, \dots, a_n) \\ \uparrow, & \text{ak } \neg P(a_1, \dots, a_n) \end{cases}$$

- (b) Charakteristická funkcia predikátu $P(x_1, \dots, x_n)$ je taká n -árna funkcia f na množine X , že pre všetky $a_1, \dots, a_n \in X$ platí:

$$f(a_1, \dots, a_n) = \begin{cases} 0, & \text{ak } P(a_1, \dots, a_n) \\ 1, & \text{ak } \neg P(a_1, \dots, a_n) \end{cases}$$

Označenie. \uparrow - nedefinovaná, \downarrow - definovaná

Veta 4.10. Ak sú $P(x_1, \dots, x_n)$, $Q(y_1, \dots, y_m)$ (primitívne) rekurzívne predikáty, sú aj

$$\begin{aligned} P(x_1, \dots, x_n) \wedge Q(y_1, \dots, y_m) \\ P(x_1, \dots, x_n) \vee Q(y_1, \dots, y_m) \\ P(x_1, \dots, x_n) \implies Q(y_1, \dots, y_m) \\ P(x_1, \dots, x_n) \iff Q(y_1, \dots, y_m) \\ \neg P(x_1, \dots, x_n) \end{aligned}$$

(primitívne) rekurzívne.

Poznámka 4.11. Vo vete 4.10 sa nežiada, aby $\{x_1, \dots, x_n\}$, $\{y_1, \dots, y_m\}$ boli disjunktné množiny premenných. Preto ak je aj $P(x_1, \dots, x_n)$ n -árny predikát a $Q(y_1, \dots, y_m)$ m -árny predikát, nemusí byť napríklad $P(x_1, \dots, x_n) \implies Q(y_1, \dots, y_m)$ $(n+m)$ -árny predikát.

Dôkaz vety 4.10: Vetu zrejmé stačí dokázať pre logické spojky \neg , \vee , lebo ostatné logické spojky už možno pomocou nich vyjadriť. Ak sú $\chi_P(x_1, \dots, x_n)$, $\chi_Q(y_1, \dots, y_m)$ charakteristické funkcie predikátorov $P(x_1, \dots, x_n)$, $Q(y_1, \dots, y_m)$, sú $\chi_P(x_1, \dots, x_n) \cdot \chi_Q(y_1, \dots, y_m)$, $1 \div \chi_P(x_1, \dots, x_n)$ charakteristické funkcie predikátorov $P(x_1, \dots, x_n) \vee Q(y_1, \dots, y_m)$, $\neg P(x_1, \dots, x_n)$. Z predpokladov vety vyplýva, že tieto charakteristické funkcie, a teda aj príslušné predikáty, sú (primitívne) rekurzívne. \square

Veta 4.12. Nech $P(x_1, \dots, x_n, z)$ je (primitívne) rekurzívny predikát, $f(x_1, \dots, x_n)$ je (primitívne) rekurzívna funkcia. Potom aj predikáty

$$(\exists z)(z \leq f(x_1, \dots, x_n) \wedge P(x_1, \dots, x_n, z)) \quad (4.12.1)$$

$$(\forall z)(z \leq f(x_1, \dots, x_n) \implies P(x_1, \dots, x_n, z)) \quad (4.12.2)$$

sú (primitívne) rekurzívne.

Dôkaz: Nech $\chi_P(x_1, \dots, x_n, z)$ je charakteristická funkcia predikátu $P(x_1, \dots, x_n, z)$. Potom funkcia $\chi(x_1, \dots, x_n) = \prod_{i=0}^{f(x_1, \dots, x_n)} \chi_P(x_1, \dots, x_n, i)$ je charakteristická funkcia predikátu (4.12.1). Pretože funkcia $\chi(x_1, \dots, x_n)$ je (primitívne) rekurzívna, je aj predikát (4.12.1) (primitívne) rekurzívny. Predikát (4.12.2) je ekvivalentný s predikátom

$$\neg(\exists z)(z \leq f(x_1, \dots, x_n) \wedge \neg P(x_1, \dots, x_n, z))$$

a teda je tiež (primitívne) rekurzívny podľa už dokázanej časti vety 4.12 a podľa vety 4.10. \square

Veta 4.14. Ak je $P(x_1, \dots, x_m)$ (primitívne) rekurzívny predikát,

$$f_1(y_{1,1}, \dots, y_{1,n_1}), \dots, f_m(y_{m,1}, \dots, y_{m,n_m})$$

sú (primitívne) rekurzívne funkcie, tak predikát $P(f_1(y_{1,1}, \dots, y_{1,n_1}), \dots, f_m(y_{m,1}, \dots, y_{m,n_m}))$ je (primitívne) rekurzívny.

Poznámka 4.15. Obdobne ako vo vete 4.10, ani tu nepredpokladáme, že premenné $y_{i,j}$, $i = 1, \dots, m$, $j = 1, \dots, n_i$ sú po dvoch rôzne.

Dôkaz vety 4.14: Ak je $\chi(x_1, \dots, x_n)$ charakteristická funkcia predikátu $P(x_1, \dots, x_n)$, tak (primitívne) rekurzívna funkcia $\chi(f_1(y_{1,1}, \dots, y_{1,n_1}), \dots, f_m(y_{m,1}, \dots, y_{m,n_m}))$ je charakteristická funkcia predikátu z vety 4.14, a teda tento predikát je (primitívne) rekurzívny. \square

Definícia. Nech $n \leq 1$. Hovoríme, že množina $A \subseteq \mathbb{N}^k$ je Σ_n -množina, ak existuje všeobecne rekurzívna relácia $R \subseteq \mathbb{N}^{k+n}$ taká, že $A = \{x_1, \dots, x_k | (\exists v_1)(\forall v_2) \dots (\frac{\exists}{\forall} v_n) R(x_1, \dots, x_k, v_1, \dots, v_n)\}$

Hovoríme, že množina $A \subseteq \mathbb{N}^k$ je typu Π_n -množina, ak existuje všeobecne rekurzívna relácia $R \subseteq \mathbb{N}^{k+n}$ taká, že, $A = \{x_1, \dots, x_k | (\forall v_1)(\exists v_2) \dots (\frac{\forall}{\exists} v_n) R(x_1, \dots, x_k, v_1, \dots, v_n)\}$

Príklad. $K = \{x | x \in W_x\} = \{x | \exists w T(x, x, w) = 0\} \text{ in } \Sigma_1$.

$$\bar{K} = \{x | x \in W_x\} = \{x | \forall w_{\text{neg}} T(x, x, w) = 0\}$$

Poznámka. $(\forall v_1)(\forall v_2)(\forall v_3)R(\bar{x}, v_1, v_2, v_3)$.

$$(\forall c(v_1, v_2, v_3))R(\bar{x}, P_1(z), P_2(z), P_3(z))$$

5. Modely vypočítateľnosti (Turingove stroje a iné modely).

Minského/registrované stroje:

Minského stroje - Registrový stroj pracuje v diskrétnom čase, to znamená, že robí jednotlivé kroky v nejakej postupnosti časových okamihov, v každom okamihu z tejto postupnosti jeden krok. Má konečnú množinu vnútorných stavov, v ktorých sa môže nachádzať; jeho činnosť v istom časovom okamihu závisí od vnútorného stavu, v ktorom sa práve nachádza. Ďalej má konečne mnoho registrov R_j , do každého z ktorých môže uložiť jedno ľubovoľne veľké prirodzené číslo. (Práve táto podmienka, uloženie ľubovoľne veľkého prirodzeného čísla do jedného registra, alebo hoci aj do celej pamäti stroja, sa nedá technicky realizovať; ľahko sa však dá dosiahnuť, že počas veľmi dlhého času sa tieto ohraničenia neprejavia.) V každom kroku svojej činnosti pracuje registrový stroj s jedným svojím registrom, ktorý závisí len od momentálneho vnútorného stavu q_i stroja. Obsah tohto registra R_j stroj zmení (najviac o jednotku) a prejde do nasledujúceho vnútorného stavu q_k . Spôsob zmeny obsahu registra R_j i vnútorný stav q_k závisí od q_i a pôvodného obsahu registra R_j . Potom sa robí ďalší krok výpočtu atď., až sa stroj dostane do takého vnútorného stavu, pri ktorom je predpísané zastavenie (resp. nie je určená ďalšia činnosť). Pred začiatkom výpočtu uložíme vstupné údaje do niektorých registrov stroja a po ukončení výpočtu čítame výsledok z určeného registra.

Definícia 5.1. (a) Symboly q_0, q_1, q_2, \dots budeme nazývať *vnútornými stavmi*. Množinu $\{q_i \mid i \in \mathbb{N}\}$ budeme značiť \overline{Q} .

(b) Symboly R_0, R_1, R_2, \dots budeme nazývať *registrami*.

Definícia 5.2. Usporiadane štvorce tvarov

$$(q_i R_j q_m q_n), \quad (q_i R_j P q_k), \quad (q_i R_j M q_k) \quad (5.2.1)$$

budeme nazývať *Minského inštrukciami*, prípadne len *inštrukciami*. (Budeme ich písaf bez čiarok.)

Definícia 5.3. Konečnú množinu Minského inštrukcií, ktorá neobsahuje dve rôzne inštrukcie s rovnakým prvkom, budeme nazývať *registrovým strojom*.

Definícia 5.4. Usporiadane dvojice tvaru

$$(q_i, (a_0, a_1, a_2, \dots)), \quad (5.4.1)$$

kde q_i je vnútorný stav a (a_0, a_1, a_2, \dots) je postupnosť prirodzených čísel, ktorá má len konečne mnoho nenulových členov, budeme nazývať *stavmi registrových strojov*, prípadne len *stavmi alebo M-stavmi*.

Definícia 5.7. Nech Z je registrový stroj. Budeme písaf

$$(q_i; a_0, a_1, a_2, \dots) \xrightarrow{Z} (q_j; b_0, b_1, b_2, \dots), \quad (5.7.1)$$

ak existuje také $n \in \mathbb{N}$, že pre všetky $x \in \mathbb{N}$, $x \neq n$ platí $a_x = b_x$ a existuje také $y \in \mathbb{N}$, že je splnená jedna z nasledujúcich podmienok:

$$(q_i R_n P q_j) \in Z \text{ a } b_n = a_n + 1 \quad (5.7.2)$$

$$(q_i R_n M q_j) \in Z \text{ a } b_n = a_n - 1 \quad (5.7.3)$$

$$(q_i R_n q_j q_y) \in Z \text{ a } b_n = a_n \neq 0 \quad (5.7.4)$$

$$(q_i R_n q_y q_j) \in Z \text{ a } b_n = a_n = 0 \quad (5.7.5)$$

Príklad 5.9. Nech $Z = \{(q_1 R_1 q_2 q_3), (q_2 R_2 M q_1), (q_3 R_0 P q_1)\}$. Podľa definície 5.3 je Z registrový stroj. Podľa definície 5.7 platí napríklad:

$$(q_1; 1, 0, 1, 0, 2) \xrightarrow{Z} (q_3; 1, 0, 1, 0, 2)$$

$$(q_1; 0, 1, 0, 0, 0) \xrightarrow{Z} (q_2; 0, 1)$$

$$(q_2; 0, 0, 1) \xrightarrow{Z} (q_1; 0, 0, 0)$$

$$(q_3; 4, 0, 3) \xrightarrow{Z} (q_1; 5, 0, 3)$$

$$(q_1; 5, 0, 3) \xrightarrow{Z} (q_3; 5, 0, 3)$$

$$(q_2; 5, 0, 3) \xrightarrow{Z} (q_1; 5, 0, 2)$$

- Definícia 5.12.** (a) Konečnú postupnosť stavov X_0, X_1, \dots, X_n nazveme *výpočtom registrového stroja Z zo stavu X_0* , ak $X_i \xrightarrow{Z} X_{i+1}$ pre všetky $i = 0, 1, \dots, n-1$ a neexistuje taký stav X_{n+1} , že $X_n \xrightarrow{Z} X_{n+1}$.
(b) Nekonečnú postupnosť stavov $X_0, X_1, \dots, X_n, \dots$ nazveme *výpočtom registrového stroja Z zo stavu X_0* , ak pre všetky $i = 0, 1, 2, \dots$ platí $X_i \xrightarrow{Z} X_{i+1}$.

- Definícia 5.13.** (a) Nech X, Y sú stavy, Z je registrový stroj. Budeme písat $X \xrightarrow{Z} Y$, ak existuje taká konečná postupnosť stavov X_0, X_1, \dots, X_n , že platí $X = X_0, Y = X_n$ a pre všetky $i = 0, 1, \dots, n-1$ platí $X_i \xrightarrow{Z} X_{i+1}$.
(b) Budeme písat $X \xrightarrow{Z} Y$, ak $X \xrightarrow{Z} Y$ a neexistuje taký stav T , že $Y \xrightarrow{Z} T$.

Definícia 5.17. Nech $n \in \mathbb{N}$, Z je registrový stroj a f je n -árna čiastočná funkcia na množine \mathbb{N} . Budeme hovoriť, že stroj Z počíta n -árnu čiastočnú funkciu f , a písat $f = \Phi_Z^n$, ak pre všetky $x_1, \dots, x_n, y \in \mathbb{N}$ platí $f(x_1, \dots, x_n) = y$ práve vtedy, keď existujú také čísla $b_1, \dots, b_k \in \mathbb{N}$, že

$$(q_1; 0, x_1, \dots, x_n) \xrightarrow{Z} (q_0; y, b_1, \dots, b_k) \quad (5.17.1)$$

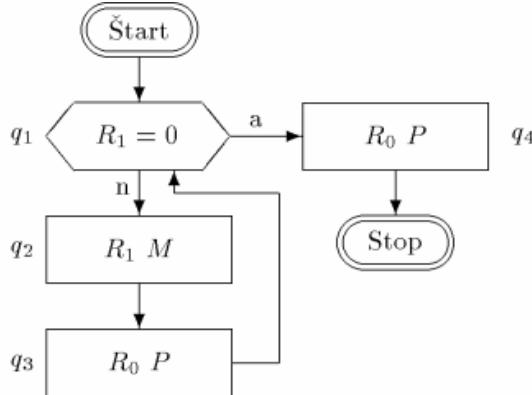
(T.j. že výpočet stroja Z zo stavu $(q_1; 0, x_1, \dots, x_n)$ sa končí stavom $(q_0; y, b_1, \dots, b_k)$.)

Veta 5.23. Ku každému registrovému stroju Z existuje taký registrový stroj Y , že pre každé $n \in \mathbb{N}$ platí $\Phi_Y^n = \Phi_Z^n$ a pre všetky $x_1, \dots, x_n \in \mathbb{N}$ je $\Phi_Y^n(x_1, \dots, x_n)$ definované práve vtedy, keď je výpočet stroja Y zo stavu $(q_1; 0, x_1, \dots, x_n)$ konečný.

Definícia 5.26. Nech je daný registrový stroj Z . Jeho *blokovú schému* dostaneme tak, že:

- (1) Každej inštrukcii stroja Z tvaru $(q_i R_j q_m q_n)$ priradíme šesťuholník, do ktorého vpíšeme „ $R_j = 0$ “.
- (2) Každej inštrukcii stroja Z tvaru $(q_i R_j M q_k)$, resp. $(q_i R_j P q_k)$ priradíme obdĺžnik, do ktorého vpíšeme „ $R_j M$ “, resp. „ $R_j P$ “.
- (3) Ak má inštrukcia, ku ktorej patrí obdĺžnik A , na štvrtom mieste rovnaký vnútorný stav ako má inštrukcia patriaca k obdĺžniku alebo šesťuholníku B na prvom mieste, spojíme A, B šípkou od A k B .
- (4) Ak má inštrukcia, ku ktorej patrí šesťuholník A , na treťom, resp. štvrtom mieste ten vnútorný stav, ktorým sa začína inštrukcia patriaca k obdĺžniku alebo šesťuholníku B , spojíme A, B šípkou od A k B ; jej začiatok pri A označíme písmenom „n“, resp. písmenom „a“ (od slov „nie“, „áno“).
- (5) Ak obsahuje stroj Z inštrukciu začínajúcu sa vnútorným stavom q_1 , doplníme blokovú schému dvojitým oválom, do ktorého vpíšeme slovo „Start“. (Takýto ovál budeme nazývať začiatocným oválom.) Potom vedieme šípku od tohto oválu k šesťuholníku alebo obdĺžniku patriacemu k tejto inštrukcii.
- (6) Ak má stroj Z vnútorný stav q_0 na treťom alebo štvrtom mieste niektornej svojej inštrukcie a nemá ho na prvom mieste žiadnej svojej inštrukcie, doplníme blokovú schému dvojitým oválom, do ktorého vpíšeme slovo „Stop“. Potom vedieme šípky od všetkých obdĺžnikov a šesťuholníkov patriacich k inštrukciám, na ktorých treťom alebo štvrtom mieste je vnútorný stav q_0 , k tomuto oválu. (Tento ovál budeme nazývať koncovým oválom.) Začiatky šípok od šesťuholníka označíme písmenami „n“ resp. „a“ podľa toho, či sa vnútorný stav q_0 nachádza na treťom alebo na štvrtom mieste príslušnej inštrukcie.
- (7) K obdĺžnikom a šesťuholníkom pripíšeme vnútorné stavy, ktorými sa začínajú k nim patriace inštrukcie.

Príklad 5.20. Stroj Z , ktorý počíta funkciu s , t.j. stroj Z s vlastnosťou $\Phi_Z^1 = s$, je napríklad $Z = \{(q_1 R_1 q_2 q_4), (q_2 R_1 M q_3), (q_3 R_0 P q_1), (q_4 R_0 P q_0)\}$.



Definícia 5.36. (Čiastočnú) funkciu f nazveme (čiastočne) vypočítateľnou na registrovom stroji, ak existuje registrový stroj Z a také číslo n , že $f = \Phi_Z^n$.

Veta 5.51. Pre každé $n \in \mathbb{N}$ a každé dva registrové stroje Z_1, Z_2 existuje taký registrový stroj Z , že čiastočná funkcia Φ_Y^{n+1} vzniká primitívou rekurziou z čiastočných funkcií $\Phi_{Z_1}^{n+2}, \Phi_{Z_2}^n$, t.j. $\Phi_Y^{n+1} = \mathcal{R}(\Phi_{Z_1}^{n+2}, \Phi_{Z_2}^n)$.

Dôkaz: Pri pomeňme najprv, že pre čiastočnú funkciu Φ_Y^{n+1} platí:

$$\begin{aligned}\Phi_Y^{n+1}(0, x_1, \dots, x_n) &= \Phi_{Z_2}^n(x_1, \dots, x_n) \\ \Phi_Y^{n+1}(y + 1, x_1, \dots, x_n) &= \Phi_{Z_1}^{n+2}(y, \Phi_Y^{n+1}(y, x_1, \dots, x_n), x_1, \dots, x_n)\end{aligned}$$

stroje Z_1, Z_2 prerobíme podľa vety 5.43 na stroje Z'_1, Z'_2 , ktoré budú počítať tie isté čiastočné funkcie $\Phi_{Z_1}^{n+2}, \Phi_{Z_2}^n$, a ktoré budú okrem toho zachovávať argumenty. Nech k je najmenší index registra, ktorý nepoužíva ani jeden zo strojov Z'_1, Z'_2 . Potom stroj Y s blokovou schémou na obrázku 5.5(a) má vo vete požadovanú vlastnosť. Vnútorné stavy v tejto blokovnej schéme doplníme (po eliminácii dvojitých obdĺžnikov) podľa cvičenia 5.35. (Ak by táto bloková schéma nebola súvislá, čo by sa mohlo stať, keby blokové schémy strojov Z_1, Z_2 neboli súvislé, vezmeme tú jej časť, ktorá obsahuje začiatkový ovál.) Potom už môžeme vypísať inštrukcie stroja Y ; pokiaľ by neboli blokovou schémou určené vnútorné stavy na tretích a štvrtých miestach niektorých inštrukcií, zvolíme ich tak, aby sa rovnali vnútornému stavu na prvom mieste inštrukcie. \square

Veta 5.55. Ku každému prirodzenému číslu n a každému registrovemu stroju Z existuje taký registrový stroj Y , že Φ_Y^n je čiastočná funkcia, ktorá vzniká z čiastočnej funkcie Φ_Z^{n+1} operáciou minimalizácie, t.j. $\Phi_Y^n = \mathcal{M}(\Phi_Z^{n+1})$.

Dôkaz: Pri pomeňme najprv, že

$$\Phi_Y^n(x_1, \dots, x_n) = \mu_y(\Phi_Z^{n+1}(y, x_1, \dots, x_n)) = 0$$

Bloková schéma stroja Y je na obrázku 5.5(b). Z' je stroj, ktorý vznikne zo Z podľa vety 5.43. Dokončenie dôkazu, ktoré je rovnaké ako pri vete 5.51, prenechávame čitateľovi. \square

Veta 5.56. Každá (čiastočne) rekurzívna funkcia je (čiastočne) vypočítateľná na registrovom stroji.

Turingove stroje:

(neboli odprednášané na Vypočítateľnosti, ale treba vedieť aspoň základ)

Turingov stroj sa skladá z troch častí: riadiacej jednotky, čítacej a zapisovacej hlavy a pásky. Riadiaca jednotka sa môže nachádzať v jednom z konečne mnohých vnútorných stavov, prijíma signály od čítacej a zapisovacej hlavy a dáva signály pre jej činnosť. Páska je lineárna a v oboch smeroch nekonečná. Je rozdelená na polička, do každého z nich sa zapisuje jedno písmeno znejakej abecedy. O poličkach pásky, do ktorých sa ešte nezapisovalo, predpokladáme, že je v nich zapísaný symbol B (z anglického „blank“); vo veľkej väčšine prípadov bude teda symbol B napísaný vo všetkých poličkach pásky až na ich konečný počet a iba výnimočne budeme uvažovať, že bol zaplnený nekonečný počet poličok pásky. Hlava číta zakaždým jedno poličko pásky, odošle signál o jeho obsahu riadiacej jednotke a po prijatí signálu od riadiacej jednotky zmení predpísaným spôsobom obsah čitaného polička a vykoná predpísaný pohyb o jedno poličko doľava alebo doprava, alebo zostane na mieste. Riadiaca jednotka v každom kroku činnosti Turingovho stroja najprv prijme signál od čítacej a zapisovacej hlavy, potom tejto hlave odošle signál pre jej činnosť a nakoniec sama prejde do nového vnútorného stavu. Ak pre niektorý vnútorný stav a pre niektorý obsah čitaného polička nie je určená ďalšia činnosť, výpočet sa ukončí.

Definícia 8.1. Usporiadane 5-tice tvarov

$$(q_i a_j a_m L q_n), \quad (q_i a_j a_m N q_n), \quad (q_i a_j a_m P q_n) \quad (8.1.1)$$

budeme nazývať *turingovskými inštrukciami*, prípadne len *inštrukciami* alebo *T-inštrukciami*. (Budeme ich písaf bez čiarok.)

Definícia 8.4. Konečnú množinu *T-inštrukcií*, ktorá neobsahuje dve rôzne inštrukcie s rovnakými prvými dvoma prvkami, budeme nazývať *Turingovým strojom*, prípadne len *T-strojom* alebo *strojom*.

Definícia 8.5. Takú konečnú postupnosť

$$a_{i_1} a_{i_2} \dots a_{i_k} q_j a_{i_{k+1}} \dots a_{i_n} \quad (8.5.1)$$

že $i_1 \neq 0, i_n \neq 0$, budeme nazývať *stavom Turingovho stroja*, prípadne len *stavom* alebo *T-stavom*.

Definícia 8.8. Nech X_1, X_2 sú *T-stavy*, T je Turingov stroj. Budeme písaf

$$X_1 \xrightarrow{T} X_2$$

ak existujú také slová v, w a také čísla $i, j, k, m, n \in \mathbb{N}$, že platí

- (a) $X_1 = v q_i a_j w, X_2 = v q_n a_m w$ a $(q_i a_j a_m N q_n) \in T$ alebo
- (b) $X_1 = v q_i a_j w, X_2 = v a_m q_n w$ a $(q_i a_j a_m P q_n) \in T$ alebo
- (c) $X_1 = v a_k q_i a_j w, X_2 = v q_n a_k a_m w$ a $(q_i a_j a_m L q_n) \in T$.

Príklad 8.12. Nech $T = \{(q_1 O I P q_1), (q_1 I O P q_1), (q_1 B B L q_2), (q_2 O O L q_2), (q_2 I I L q_2), (q_2 B B P q_3), (q_3 O O N q_0), (q_3 I I N q_0)\}$. Podľa definície 8.4 je T Turingov stroj. Podľa definície 8.8 platí napríklad:

$$\begin{array}{ll} q_1 O I I O \xrightarrow{T} I q_1 I I O & I O q_2 O I \xrightarrow{T} I q_2 O O I \\ I q_1 I I O \xrightarrow{T} I O q_1 I O & I q_2 O O I \xrightarrow{T} q_2 I O O I \\ I O q_1 I O \xrightarrow{T} I O O q_1 O & q_2 I O O I \xrightarrow{T} q_2 B I O O I \\ I O O q_1 O \xrightarrow{T} I O O I q_1 & q_2 B I O O I \xrightarrow{T} q_3 I O O I \\ I O O I q_1 \xrightarrow{T} I O O q_2 I & q_3 I O O I \xrightarrow{T} q_0 I O O I \\ I O O q_2 I \xrightarrow{T} I O q_2 O I & \end{array}$$

Definícia 8.13. (a) Konečnú postupnosť *T-stavov* X_0, X_1, \dots, X_n nazveme *výpočtom Turingovho stroja T zo stavu X₀*, ak $X_i \xrightarrow{T} X_{i+1}$ pre všetky $i = 0, 1, \dots, n-1$ a neexistuje taký stav X_{n+1} , že $X_n \xrightarrow{T} X_{n+1}$.

(b) Nekonečnú postupnosť *T-stavov* X_0, X_1, X_2, \dots nazveme *výpočtom Turingovho stroja T zo stavu X₀*, ak pre všetky $i = 0, 1, 2, \dots$ platí $X_i \xrightarrow{T} X_{i+1}$.

Definícia 8.14. (a) Nech X, Y sú *T-stavy*, T je Turingov stroj. Budeme písaf $X \xrightarrow{T} Y$, ak existuje taká konečná postupnosť *T-stavov* X_0, X_1, \dots, X_n , že platí $X = X_0, Y = X_n$ a pre všetky $i = 0, 1, \dots, n-1$ platí $X_i \xrightarrow{T} X_{i+1}$.

(b) Budeme písaf $X \xrightarrow{T} Y$, ak $X \xrightarrow{T} Y$ a neexistuje taký stav Z , že $Y \xrightarrow{T} Z$.

Definícia 8.20. Nech T je Turingov stroj a A je jeho abeceda. Symbolom Rez_T označíme také čiastočné zobrazenie množiny A^* do množiny A^* , že pre všetky $v, w \in A^*$ platí $\text{Rez}_T(v) = w$ práve vtedy, keď $q_1 v \xrightarrow{T} q_0 w$.

Definícia 8.24. (a) Nech T je Turingov stroj a n prirodzené číslo. Znakom Φ_T^n budeme označovať takú n -árnu čiastočnú funkciu f na množine \mathbb{N} , že pre všetky $y, x_1, \dots, x_n \in \mathbb{N}$ platí $f(x_1, \dots, x_n) = y$ práve vtedy, keď $\text{Rez}_T(OI^{x_1}OI^{x_2}O\dots OI^{x_n}O) = OI^yO$. Čiastočnú funkciu Φ_T^n budeme nazývať *n-árnou čiastočnou funkciou*, ktorú počíta stroj T .
(b) Budeme hovoriť, že n -árna (čiastočná) funkcia f je (čiastočne) vypočítateľná na Turingovom stroji (alebo len (čiastočne) T -vypočítateľná), ak existuje taký Turingov stroj T , že $f = \Phi_T^n$.

Príklad 8.25. Nech $T = \{(q_1 OIL q_2), (q_2 BIL q_3), (q_3 BIL q_4), (q_4 BON q_0)\}$. Pre ľubovoľné prirodzené číslo k platí $q_1 OI^k O \xrightarrow{T} q_2 BI^{k+1} O \xrightarrow{T} q_3 BI^{k+2} O \rightarrow q_4 BI^{k+3} O \xrightarrow{T} q_0 OI^{k+3} O$, a teda $\text{Rez}_T(OI^k O) = OI^{k+3} O$. Preto $\Phi_T^1(k) = k + 3$ pre všetky $k \in \mathbb{N}$.

Veta 8.53. Každá (čiastočne) M -vypočítateľná funkcia je (čiastočne) T -vypočítateľná.

Veta 8.54. Každá (čiastočne) rekurzívna funkcia je (čiastočne) vypočítateľná na Turingovom stroji.

6. Ekvivalentnosť modelov vypočítateľnosti.

Ekvivalentnosť M-vypočítateľnosti a rekurzívnosti:

(Kódovanie pre oba typy ekvivalencie je uvedené v 9.)

Označenie 6.5. Ak k je číslom M -stroja Z , budeme čiastočnú funkciu Φ_Z^n označovať aj symbolom Φ_k^n . Ak k nebude číslom stroja, budeme klásť $\Phi_k^n = \emptyset$ (t.j. nikde nedefinovanej čiastočnej funkci).

Definícia 6.11. (a) Symbolom $M\text{prech}$ (z, x, y) označíme predikát „stav číslo y vzniká zo stavu číslo x jediným krokom výpočtu stroja číslo z “.
 (b) Pre každé $n \in \mathbb{N}$ symbolom $Mv\text{ýp}^{n+2}(y, z, x_1, \dots, x_n)$ označíme predikát „ y je číslom výpočtu stroja číslo z zo stavu $(q_1; 0, x_1, \dots, x_n)$, pričom tento výpočet končí vnútorným stavom q_0 “.
 (c) Symbolom $Mv\text{ýp}^{n+2}$ označíme charakteristickú funkciu predikátu $Mv\text{ýp}^{n+2}$.
 (d) Symbolom $mobs(y)$ označíme obsah registra R_0 v poslednom člene konečnej postupnosti stavov, ktorá má číslo y .

Veta 6.13. (a) Predikát $M\text{prech}$ a predikáty $Mv\text{ýp}^{n+2}$ ($n \in \mathbb{N}$) sú primitívne rekurzívne.

(b) Funkcia $mobs$ a funkcie $Mv\text{ýp}^{n+2}$ ($n \in \mathbb{N}$) sú primitívne rekurzívne.

Dôkaz: Primitívna rekurzívnosť predikátu $M\text{prech}$ vyplýva z vyjadrenia

$$\begin{aligned} M\text{prech}(z, x, y) &\iff \\ &\iff (\exists i, j, k, m \leq x + y + z) \\ &\quad (\text{ex}(0, x) = i \wedge \text{ex}(0, y) = k \wedge x > 0 \wedge y > 0 \wedge \\ &\quad \wedge (\forall t \leq x + y) (t \neq j \implies \text{ex}(t + 1, x) = \text{ex}(t + 1, y)) \wedge \\ &\quad \wedge ((\text{ex}(i, z) = 2c^3(j, k, m) + 2 \wedge \\ &\quad \wedge \text{ex}(j + 1, x) = \text{ex}(j + 1, y) \wedge \text{ex}(j + 1, x) > 0) \vee \\ &\quad \vee (\text{ex}(i, z) = 2c^3(j, m, k) + 2 \wedge \\ &\quad \wedge \text{ex}(j + 1, x) = 0 \wedge \text{ex}(j + 1, y) = 0) \vee \\ &\quad \vee (\text{ex}(i, z) = 4c(j, k) + 1 \wedge \text{ex}(j + 1, x) \dot{-} 1 = \text{ex}(j + 1, y)) \vee \\ &\quad \vee (\text{ex}(i, z) = 4c(j, k) + 3 \wedge \text{ex}(j + 1, x) + 1 = \text{ex}(j + 1, y))) \end{aligned} \tag{6.13.1}$$

Na vysvetlenie uvedme, že posledných šesť riadkov po rade zodpovedá tomu, že vykonávaná inštrukcia je $(q_i R_j q_k q_m)$, $(q_i R_j q_m q_k)$, $(q_i R_j M q_k)$, resp. $(q_i R_j P q_k)$; prechádza sa z vnútorného stavu q_i do vnútorného stavu q_k .

Pre každé $n \in \mathbb{N}$ teraz primitívna rekurzívnosť predikátu $Mv\text{ýp}^{n+2}$ vyplýva z vyjadrenia

$$\begin{aligned} Mv\text{ýp}^{n+2}(y, z, x_1, \dots, x_n) &\iff \\ &\iff (\text{ex}(0, y) = 2^1 \cdot 3^0 \cdot 5^{x_1} \cdot 7^{x_2} \cdots p_{n+1}^{x_n} \wedge \\ &\quad \wedge (\forall t \leq \text{npr}(y) \dot{-} 2) M\text{prech}(z, \text{ex}(t, y), \text{ex}(t + 1, y)) \wedge \\ &\quad \wedge \text{ex}(0, \text{ex}(\text{npr}(y) \dot{-} 1, y)) = 0 \wedge \text{ex}(0, z) = 0) \end{aligned} \tag{6.13.2}$$

Na vysvetlenie uvedme, že prvý riadok pravej strany vyjadruje, že výpočet začína predpísaným stavom; n je pevné a tri bodky teda možno nahradí konkrétnym rozpisom, napr. pre $n = 0$ vyjdé $\text{ex}(0, y) = 2$. Druhý riadok vyjadruje, že y je číslom počiatočného úseku nejakého výpočtu. Tretí riadok vyjadruje, že v poslednom člene tohto úseku je vnútorný stav q_0 a že stroj s číslom z neobsahuje inštrukciu začínajúcu q_0 .

Primitívna rekurzívnosť funkcie $mobs$ vyplýva z vyjadrenia (ktoré zoberieme ďalej ako definíciu)

$$mobs(y) = \text{ex}(1, \text{ex}(\text{npr}(y) \dot{-} 1, y)) \tag{6.13.3}$$

Teda: berie sa exponent posledného člena v rozklade y na prvočinitele, ten sa znova rozloží na prvočinitele a berie sa exponent prvočísla $p_1 = 3$.

Primitívna rekurzívnosť funkcií $Mv\text{ýp}^{n+2}$ bezprostredne vyplýva z primitívnej rekurzívnosti predikátov $Mv\text{ýp}^{n+2}$. \square

Veta 6.17. (Čiastočná) funkcia f je (čiastočne) rekurzívna práve vtedy, keď je (čiastočne) M -vypočítateľná.

Dôkaz: Znenie so slovami v zátvorkách dostávame bezprostredne z viet 6.8 a 6.15. Pre znenie bez týchto slov uvažujeme (totálnu) n -árnu funkciu f . Ak je funkcia f rekurzívna, tak $f = \Phi_Z^n$ pre nejaký M -stroj Z a f je M -vypočítateľná. Obrátene, ak f je M -vypočítateľná, tak existuje také a , pre ktoré platí (6.15.1), a potom aj

$$f(x_1, \dots, x_n) = \text{mobs}(\mu_y(\text{mvp}_a^{n+1}(y, x_1, \dots, x_n) = 0)) \quad (6.17.1)$$

Funkciu f sme teda vyjadrili pomocou skladania rekurzívnych funkcií a regulárnej minimalizácie. Teda f je rekurzívna funkcia, čo bolo treba dokázať. \square

Definícia. (recursive many-to-one reduction) Nech $A, B \subseteq \mathbb{N}$ sú rekurzívne očíslovateľné (spočitatelné) množiny. Hovoríme, že množina A je m -redukovaná (many-to-one) k množine B a píšeme $A \leq_m B \Leftrightarrow$ existuje všeobecne rekurzívna funkcia f taká, že pre všetky $x \in \mathbb{N}$ platí $x \in A \Leftrightarrow f(x) \in B$.

Množiny A, B sú m -ekvivalentné (píšeme $A =_m B$ – trojité rovná sa) vtedy a len vtedy, ak $A \leq_m B$ a $B \leq_m A$.

Veta 6.1.1 Relácia \leq_m je reflexívna a tranzitívna na $P(\mathbb{N})$. Relácia $=$ – trojité – je reláciou ekvivalencie na $P(\mathbb{N})$.

Definícia. Množina $A \subseteq \mathbb{N}$ je kreatívna (tvorivá), vtedy a len vtedy, ak A je rekurzívne očíslovateľná a $(\forall B)(B \in ROM \Rightarrow B \leq_m A)$

Veta 7.1.1 Množina $K_R = \{c(y, x) | y, x \in \text{Dom}(m_{univ}^2)\}$ je kreatívna. (Doména je Arg).

Poznámka. $m_{univ}^2(y, x) = (\phi_y(x)$ je definovaná).

Dôkaz. a) K_R je rekurzívne očíslovateľná množina, lebo je definičným oborom nejakej čiastočne rekurzívne funkcie. Ktorej? $g(z) = m_{univ}^2(L(z), R(z))$.

b) ukážeme $(\forall B)(V \in ROM \Rightarrow \leq_m K_R)$ To znamená, že $\text{i}\chi_B(x) = 0$ ak $x \in B$ a je nedefinovaná ak $x \notin B$

$$\text{i}\chi_B(x) = m_{univ}^2(b, x)$$

$$x \in B \Leftrightarrow \text{i}\chi_B(x) = 0 \Leftrightarrow m_{univ}^2(b, x) = 0 \Leftrightarrow (b, x) \in \text{Dom}(m_{univ}^2) \Leftrightarrow c(b, x) \in K_R$$

Veta 6.45. Množina

$$K_0 = \{y \mid y \in \text{Dom}(\text{muniv}^1)\} \quad (6.45.1)$$

je kreatívna.

Dôkaz: Ku každému M -stroju Z a ku každému $a \in \mathbb{N}$ zostrojíme stroj $S(Z, a)$ takto: Najprv zvýšime indexy všetkých vnútorných stavov stroja Z okrem stavu q_0 o číslo a ; tým dostaneme M -stroj Z' . Potom pridáme k stroju Z' a inštrukcií $(q_1 R_1 P q_2), (q_2 R_1 P q_3), \dots, (q_a R_1 P q_{a+1})$. Zrejme pre každý stroj Z a každé a platí

$$\Phi_Z^1(a) = \Phi_{S(Z, a)}^0 \quad (6.45.2)$$

Označme teraz g tú binárnu funkciu, ktorá každému číslu $y \in \mathbb{N} \setminus \{0\}$ a každému $a \in \mathbb{N}$ priraďuje číslo stroja $S(Z, a)$, kde Z je stroj s číslom y ; nech ďalej $g(0, a) = 0$. Funkcia g je rekurzívna. Dokážeme teraz, že $K \leq_m K_0$. Nech $z \in K$, $z = c(y, x)$ a nech Z je stroj s číslom y . Potom $z \in K$ práve vtedy, keď $\Phi_Z^1(x)$ je definovaná, t.j. keď $\Phi_{S(Z, x)}^0$ je definovaná, t.j. keď $g(y, x) \in K_0$. Teda platí

$$z \in K \Leftrightarrow g(l(z), r(z)) \in K_0 \quad (6.45.3)$$

a funkcia $f(z) = g(l(z), r(z))$ je rekurzívna. Množina K_0 je zrejme rekurzívne spočitatelná, a teda podľa lemy 6.44 kreatívna. \square

Veta 7.1.2 1. Kreatívna množina nie je všeobecne rekurzívna (jej doplnok, komplement, nie je generovatený).

2. A je kreatívna, $B \in ROM$, , $A \leq_m B \Rightarrow B$ je kreatívny.
3. Ak A a B sú kreatívne, tak $A =_M B$.
4. kreatívne množiny tvoria triedu ekvivalencie.

Dôkaz.

1. Ak by kreatívna množina bola všeobecne rekurzívnu, potom podľa (g) sú všetky ROM všeobecne rekurzívne, čo je spor.
2. $(\forall \alpha)(\alpha \leq_m A \text{ AA je kreatívna, a } A \leq_m B, B \text{ je } ROM, \text{ z tranzitívnosti vyplýva, že } (\forall \alpha)(\alpha \in ROM \Rightarrow \alpha \leq_n B))$
3. $A \leq_m B \wedge B \leq_M A$ a teda $A =_m B$
4. vyplýva priamo z (3)

Definícia. $K = \{x | x \in W_x\} - \phi_x(x)$ je definovaná
 $K_0 = \{c(x, v) | v \in W_x\} - \phi_x(v)$ je definovaná

Veta 7.2.1 Množiny K a K_0 sú rekurzívne očíslovateľné a nie sú všeobecne rekurzívne.

Dôkaz. K, K_0 sú rekurzívne očíslovateľné, lebo sú doménou nejakej CRF .

a) K je $VRM \rightarrow \bar{K}$ je $VRM \rightarrow \bar{K}$ je ROM . Ukážeme že $\bar{K} \notin ROM$.

Nech \bar{K} je rekurzívne očíslovateľná, potom má číslo, $(\exists e)\bar{K} = W_e$.

$$\bar{K} = \{x | x \in W_x\} = W_e.$$

Nech $e \in \bar{K} = W_e \Rightarrow e \in W_e \Rightarrow e \notin W_e$.

Nech $e \notin \bar{K} = W_e \Rightarrow e \notin W_e \Rightarrow e \in K \Rightarrow e \in W_e$. Spor – Russelov paradox.

A teda K nie je VRM

b) Redukciou. $x \in K \Leftarrow c(x, x) \in K_0 \Rightarrow c_{\chi_k}(x) = 0 \Leftrightarrow_{K_0} (c(x, x)) = 0$.

Ekvivalentnosť T-vypočítateľnosti a rekurzívnosti:

(nebolo odprednášané, ale je to v skriptách)

Definícia 9.10. (a) $Tstv(x)$ bude znamenať „existuje T -stav s číslom x “, inými slovami „ x je číslom nejakého T -stavu“.

(b) $Prech(x, y, z)$ bude znamenať „existujú také T -stavy X, Y s číslami x, y a taký T -stroj Z s číslom z , že $X \xrightarrow{Z} Y$ “.

(c) $Tvýp(x, y, z)$ bude znamenať „ x je číslo výpočtu T -stroja číslo y zo stavu číslo z “.

(d) $Tvýp_0(x, y, z)$ bude znamenať „ $Tvýp(x, y, z)$ a posledný člen vo výpočte číslo x sa začína vnútorným stavom q_0 “.

(e) $tvst^n(x_1, \dots, x_n)$ bude pre ľubovoľné $n \in \mathbb{N}$ a všetky x_1, \dots, x_n znamenať číslo stavu

$$q_1 O I^{x_1} O I^{x_2} O \dots O I^{x_n} O$$

(f) $tvýp_0$ bude charakteristickou funkciou predikátu $Tvýp_0$.

(g) $tobs(x)$ bude znamenať počet znakov I v poslednom člene konečnej postupnosti T -stavov, ktorá má číslo x ; ak taká postupnosť neexistuje, potom $tobs(x) = 0$.

Veta 9.21. Pre každú (čiastočnú) funkciu f na množine \mathbb{N} sú následujúce podmienky ekvivalentné:

- (a) f je (čiastočne) M -vypočítateľná;
- (b) f je (čiastočne) T -vypočítateľná;
- (c) f je (čiastočne) rekurzívna.

Definícia 9.22. Množinu slov nazveme *S-množinou*, ak existuje taká konečná podmnožina A množiny $\overline{A} \setminus \{B\}$, že $X \subseteq A^*$.

Jedným z hlavných dôvodov, pre ktorý sa obmedzíme na *S-množiny*, je, že v praktických príkladoch sa takéto množiny slov vyskytujú najčastejšie. Niektoré ďalšie dôvody uvedieme neskôr.

Definícia 9.26. (a) *S-množinu X budeme nazývať (primitívne) rekurzívnu*, ak je množina $\text{Num}(X)$ (primitívne) rekurzívna.

(b) *S-množinu X budeme nazývať rekurzívne spočítateľnou*, ak je množina $\text{Num}(X)$ rekurzívne spočítateľná.

Dôkaz nasledujúcich troch viet ponechávame čitateľovi.

Veta 9.27. (a) Množinové zjednotenie, prienik a rozdiel (primitívne) rekurzívnych *S-množín* sú (primitívne) rekurzívne *S-množiny*.

(b) Množinové zjednotenie a prienik rekurzívne spočítateľných *S-množín* sú rekurzívne spočítateľné *S-množiny*.

Veta 9.28. (a) Existuje rekurzívna *S-množina*, ktorá nie je primitívne rekurzívna.

(b) Existuje rekurzívne spočítateľná *S-množina*, ktorá nie je rekurzívna.

Veta 9.29. Nech X je rekurzívna *S-množina*, $Y \subseteq X$ a $Y, X \setminus Y$ sú rekurzívne spočítateľné *S-množiny*. Potom Y je rekurzívna *S-množina*.

7. Churchova téza.

My sme sa zaoberali hlavne algoritmami na počítanie čiastočných funkcií na množine \mathbb{N} a preskúmali sme tieto tri spresnenia pojmu algoritmicky (čiastočne) vypočítateľných funkcií:

- (čiastočne) rekurzívne funkcie
- (čiastočne) M -vypočítateľné funkcie
- (čiastočne) T -vypočítateľné funkcie

Vzhľadom na to, že vo všetkých troch prípadoch ide o spresnenie toho istého pojmu, neprekvapuje nás, že tieto tri pojmy majú mnoho spoločných vlastností. Videli sme však, že tieto pojmy majú spoločné dokonca všetky vlastnosti, že sú ekvivalentné (pozri vetu 9.21). Matematici preskúmali mnoho ďalších spresnení pojmu algoritmicky (čiastočne) vypočítateľných funkcií, a vo všetkých prípadoch sa ukázalo, že tieto pojmy sú ekvivalentné s našimi troma pojмami. To viedlo rôznych matematikov k vysloveniu nasledujúcej hypotézy:

10.6. Churchova téza: Systém všetkých algoritmicky (čiastočne) vypočítateľných funkcií na množine \mathbb{N} je totožný so systémom všetkých (čiastočne) rekurzívnych funkcií (na množine \mathbb{N}).

Churchovu tézu nie je možné matematicky dokázať; nie je totiž presným matematickým tvrdením, nakolko sa v nej vyskytuje pojem algoritmickej (čiastočnej) vypočítateľnosti, ktorý nie je ani presne definovaný, ani nemáme axiómy opisujúce jeho vzťah k iným pojmom. Pri pokusoch o takúto definíciu alebo takéto axiómy už obvykle ohraničíme triedu všetkých algoritmov a opíšeme iba algoritmy istého typu. Takto teda dostaneme len (možno nový) špeciálny pojem algoritmickej (čiastočnej) vypočítateľnosti, o ktorom sa obvykle bez principiálnych fažkostí dá dokázať, že je ekvivalentný s doterajšími pojмami. To môže byť ďalší dôvod pre uznanie Churchovej tézy, nie však jej matematický dôkaz.

Napriek tomu, že Churchovu tézu nie je možné matematicky dokázať, a teda ani používať v dôkazoch matematických viet, môže nám byť táto hypotéza pri dôkazoch viet z teórie algoritmov užitočná. Ak sa nám totiž podarí zdôvodniť platnosť nejakej matematickej vety použitím Churchovej tézy, je obvykle už len vecou rutiny prepracovať toto zdôvodnenie na presný matematický dôkaz. Napríklad ak pomocou Churchovej tézy dokážeme existenciu nejakého algoritmu, tak už obvykle bez principiálnych fažkostí dokážeme existenciu Turingovho stroja, ktorý simuluje prácu tohto algoritmu. Pritom používanie Churchovej tézy nám často umožňuje vyhnúť sa nepríjemným technickým detailom, ktoré zneprehľadňujú presné dôkazy.

Budeme teraz charakterizovať niektoré pojmy týkajúce sa čiastočných funkcií na množine \mathbb{N} a podmnožín množín \mathbb{N} a \mathbb{N}^n pomocou Churchovej tézy; správnosť týchto charakterizácií necháme na rozmyslenie čitateľovi. Znova upozorňujeme, že nejde o presné matematické vety. Ide však o praxou (v širšom zmysle) overené tvrdenia.

- Tvrdenie 10.8.** (a) n -árna funkcia f na \mathbb{N} je rekurzívna práve vtedy, keď existuje algoritmus, ktorý každej n -tici $(x_1, \dots, x_n) \in \mathbb{N}^n$ priradí hodnotu $f(x_1, \dots, x_n)$.
- (b) n -árna čiastočná funkcia f je čiastočne rekurzívna práve vtedy, keď existuje algoritmus, ktorý každej n -tici $(x_1, \dots, x_n) \in \mathbb{N}^n$ priradí hodnotu $f(x_1, \dots, x_n)$, pokiaľ je $f(x_1, \dots, x_n)$ definované, a ktorý sa nekončí, ak $f(x_1, \dots, x_n)$ nie je definované.
- (c) Množina $M \subseteq \mathbb{N}^n$ je rekurzívna práve vtedy, keď existuje algoritmus, ktorý pre každé $(x_1, \dots, x_n) \in \mathbb{N}^n$ dá odpoved „áno“, ak $(x_1, \dots, x_n) \in M$ a dá odpoved „nie“, ak $(x_1, \dots, x_n) \notin M$.
- (d) Množina $M \subseteq \mathbb{N}^n$ je rekurzívne spočítateľná práve vtedy, keď existuje algoritmus, ktorý pre každé $(x_1, \dots, x_n) \in \mathbb{N}^n$ dá odpoved „áno“, ak $(x_1, \dots, x_n) \in M$ a nedá žiadnu odpoved, ak $(x_1, \dots, x_n) \notin M$.

Tvrdenie 10.10. Množina $M \subseteq \mathbb{N}^n$ je rekurzívne spočítateľná práve vtedy, keď existuje generujúci algoritmus, ktorý generuje množinu M .

8. Problém zastavenia a iné nerozhodnutelné problémy.

Diagonalizácia totálnych funkcií:

Snažíme formalizmom zachytiť všetky (totálne) intuitívne algoritmicky vypočítateľné funkcie.

Diagonálizácia pre funkcie jednej premennej umožňuje nájsť totálnu funkciu, ktorá nie je opísaná v rámci daného formalizmu.

Máme generátor postupností, a z nich vyberieme syntakticky správne programy (P_0, \dots, P_m). Pre každý sa spýtame, či počíta funkciu jednej premennej. Teraz máme programy (P_0, \dots, P_s), kde program P_i počíta funkciu f_i .

$\Sigma = \{a_0, \dots, a_m\}$	Σ^*	Σ^2	Σ^P
generare $a_0, \dots, a_m a_0a_1, \dots, a_ma_1$	\downarrow	\downarrow	\downarrow refacere je program
	P_0	P_1	P_m
	P_0		pozitia finala. 1 prelungire
			P_m

Dokazujeme sporom: Nech $h(x) = f_x(x) + 1$ (zmeníme hodnotu, funkcia je), nech $h(x)$ má program $P_a \rightarrow h(x) = f_a(x)$:

$f_a(x) + 1 = f_a(x) \rightarrow$ (za x dosadíme a) $f_a(a) + 1 = f_a(a)$, spor – sú definované a majú rôzne hodnoty.

Funkcia $h(x)$ bola totálna intuitívne vypočítateľná, ale naša formalizácia ju nezachytila.

Z tohto dôvodu sa snažíme formalizovať čiastočne definované funkcie, kde nám nestane spor pri diagonalizácii, obe funkcie budú nedefinované a budú sa rovnat.

Veta 1.4.1 Existuje presne \aleph_0 intuitívne algoritmicky vypočítateľných čiastočne funkcií. Existuje presne \aleph_0 intuitívne algoritmické vypočítateľných totálnych funkcií.

Dôkaz. a) $\lambda_X(0), \lambda_X(1), \dots, \aleph_0$ totálnych.

b) je maximálne \aleph_0 programov a teda vypočítateľných čiastočne je najviac \aleph_0 . A navyše totálne sú podmnožinou čiastočných.

Po a) máme ohraničenie zdola, po b) ohraničenie zhora: $\aleph_0 \leq |\text{totálne}| \leq |\text{čiastočné}| \leq \aleph_0$
 Označenie. \aleph_0 – alef 0

Veta 2.0.2 Existujú nevypočítateľné funkcie.

Dôkaz. Ak je vypočítateľná, tak existuje program, programov je spočítateľne veľa, ale funkciu jednej premennej z N do $\{0, 1\}$ je nespočítateľne veľa.

Veta 2.0.3 Každá intuitívne algoritmicky vypočítateľná vypočítateľná čiastočne funkcia má \aleph_0 programov.

Dôkaz. Máme funkciu, pridám $x = x + 1, x = x - 1$

Veta 2.0.4 Neexistuje totálne algoritmicky vypočítateľná funkcia g , aby pre všetky $x, y: g(x, y) = 0$.
 AK $\phi_x(y)$ nie je definovaná, a 1 ináč.

Dôkaz. Sporom: Diagonalizácia – $g(x, y)$ - univerzálne rozhodovacia pre pre naše algoritmy. $\Psi(x) = 1$ ak $g(x, x) = 0$ ($\phi_x(x)$ nie je definovaná) a je nedefinovaná ak $g(x, x) = 1$ ($\phi_x(x)$ je definovaná). $\Psi(x)$ má svoje číslo, $\Psi(x) = \phi_a(x)$. Zoberiem $\phi_a(z) = \Psi(a)$ je definovaná, práve vtedy ak $g(a, a) = 0$ ale to práve vtedy keď $\phi_a(a)$ nie je definovaná.

Poznámka. Z dôkazu vyplýva, že funkcia je buď totálna alebo vypočítateľná.

(Šuster mi odporučil vedieť práve tieto dve vety)

Definícia 6.8.4 *Hovoríme, že problém je turingovsky riešiteľný, ak existuje DTS, ktorý zastaví na každom vstupe a rieši daný problém.*

Téza 6.8.5 (Church-Turing) *Turingovská riešiteľnosť je ekvivalentná algoritmickej riešiteľnosti.*

Definícia 6.8.5 *Hovoríme, že problém je rozhodnuteľný, ak je to riešiteľný rozhodovací problém.*

Definícia 6.8.6 *Ak rozhodovací problém nie je rozhodnuteľný, hovoríme, že je nerozhodnuteľný.*

Definícia 6.8.7 *Hovoríme, že problém je čiastočne rozhodnuteľný, ak je to rozhodovací problém a jeho zakódovaním do jazyka dostaneme rekurzívne vyčísliteľný jazyk.*

Definícia 6.11.5 *Vlastnosť S jazykov z triedy \mathcal{L} sa nazýva netriviálna, ak existujú $L_1, L_2 \in \mathcal{L}$ také, že $L_1 \in S$ a $L_2 \notin S$. V opačnom prípade hovoríme, že S je triviálna.*

Veta 6.11.5 (Rice) *Každá netriviálna vlastnosť rekurzívne vyčísliteľných jazykov je nerozhodnuteľná.*

Dôkaz. Redukciou na univerzálny jazyk. Nech S je netriviálna vlastnosť \mathcal{L}_{RE} , ktorú vieme rozhodovať. BUNV nech $\emptyset \notin S$ (inak zoberieme komplement S , tiež netriviálnu vlastnosť, ktorú tiež vieme rozhodnúť). Nech $L \in S$ je ľubovoľný jazyk. Taký jazyk určite existuje a existuje preň TS A_L . Na základe týchto predpokladov zostrojme DTS pre L_U , ktorý na každom vstupe zastane.

Náš stroj dostane na vstupe $\langle A \rangle$ a w . Z nich zostrojí kód takého TS A_w , ktorý pracuje nasledovne: Vstupné slovo x , ktoré dostane, si odloží na jednu pásku. Na druhú si napiše slovo w (ktoré má „zadrôtované“ v prechodovej funkcií) a simuluje na ňom A . Ak A slovo w akceptuje, vráti sa k pôvodnému vstupu a simuluje na ňom TS A_L . (Uvedomte si, že keď poznáme $\langle A \rangle$, $\langle A_L \rangle$ a w , vieme algoritmicke zstrojiť náš A_w .)

Rozoberme dva prípady: Ak $w \in L(A)$, tak A_w akceptuje x práve vtedy, keď ho akceptuje A_L . Preto $L(A_w) = L$. Ak $w \notin L(A)$, tak A_w sa k simulácii A_L nikdy nedostane, preto $A_w = \emptyset$.

My vieme rozhodovať vlastnosť S . Rozhodneme preto, či nami zstrojený A_w má vlastnosť S . Ak dostaneme odpoveď áno, tak $L(A_w) = L$, a teda $w \in L(A)$. Ak dostaneme odpoveď nie, tak $L(A_w) = \emptyset$ a $w \notin L(A)$. Tým sme teda rozhodli, či $w \in L(A)$. To ale nevieme, preto nemôžeme vedieť rozhodovať ani S .

Dôsledok 6.11.6 *Prázdnosť, konečnosť, regulárnosť, bezkontextovosť, atď. rekurzívne vyčísliteľných jazykov je nerozhodnuteľná. Každá z týchto vlastností je totiž netriviálna (máme konečné aj nekonečné jazyky, prázdne aj neprázdne, ...).*

Veta 6.11.7 (Rice) Vlastnosť S rekurzívne vyčísliteľných jazykov je čiastočne rozhodnutelná práve tedy, keď sú splnené nasledovné tri vlastnosti:

1. Ak $L \in S$, tak aj každý $L' \in \mathcal{L}_{RE}$, ktorý je nadmnožinou L , má vlastnosť S .
2. Ak $L \in S$, tak existuje konečná podmnožina L , ktorá má vlastnosť S .
3. Množina konečných jazykov v S je rekurzívne vyčísliteľná.

Dôkaz.

- Ak neplatí 1, tak $L_S \notin \mathcal{L}_{RE}$. Sporom. Nech $L_S \in \mathcal{L}_{RE}$, nech A_S je TS, ktorý ho akceptuje. Ďalej nech $L_1 \subseteq L_2$ sú rekurzívne vyčísliteľné jazyky, $L_1 \in S$, $L_2 \notin S$. Nech A_1 , A_2 sú TS pre L_1 , L_2 . Zostrojíme TS pre L_U^C , čo bude hľadaný spor.

Náš stroj na vstupe dostane $\langle A \rangle$ a w . Z nich zostrojí kód TS A_w , ktorý bude fungovať nasledovne: Ak $w \in L(A)$, tak $L(A_w)$ bude L_2 , inak L_1 . Spustíme A_S na $\langle A_w \rangle$. Ten akceptuje iff $L(A_w)$ má vlastnosť S , teda $L(A_w) = L_1$, teda $w \notin L(A)$. Teda náš stroj akceptuje práve komplement univerzálneho jazyka, čo je hľadaný spor.

Ukážeme ešte, ako bude vyzerať (t.j. ako zostrojiť kód) A_w s požadovanou vlastnosťou. A_w dostane vstup x . Naraz bude simulovať: A_1 na slove x , A_2 na slove x a A na slove w . Slová z L_1 máme vždy akceptovať, preto ak A_1 akceptuje, aj A_w akceptuje. Navyše ak časom zistíme, že A akceptoval w , máme akceptovať aj tie slová, ktoré sú v $L_2 \setminus L_1$. Preto ak A akceptoval w , akceptujeme aj tie x , ktoré akceptoval A_2 .

- Ak neplatí 2, tak $L_S \notin \mathcal{L}_{RE}$. Sporom. Nech $L_S \in \mathcal{L}_{RE}$, nech A_S je TS, ktorý ho akceptuje. Ďalej nech $L \in S$, ale žiadna konečná podmnožina L nie je v S . Nech A_L je TS pre L . Zostrojíme TS pre L_U^C , čo bude hľadaný spor.

Náš stroj na vstupe dostane $\langle A \rangle$ a w . Z nich zostrojí kód TS A_w , ktorý bude fungovať nasledovne: Ak $w \in L(A)$, tak $L(A_w)$ bude nejaká konečná podmnožina L , inak $L(A_w)$ bude L . Spustíme A_S na $\langle A_w \rangle$. Ten akceptuje iff $L(A_w)$ má vlastnosť S , teda $L(A_w) = L$, teda $w \notin L(A)$. Teda náš stroj akceptuje práve komplement univerzálneho jazyka, čo je hľadaný spor.

Ukážeme ešte, ako bude vyzerať (t.j. ako zostrojiť kód) A_w s požadovanou vlastnosťou. A_w dostane vstup x . Naraz bude simulovať A_L na slove x a A na slove w . Ak A_L akceptuje, aj A_w akceptuje. Ak A akceptuje, A_w sa zasekne. Zjavne ak $w \notin L(A)$, tak $L(A_w) = L$. Ak $w \in L(A)$, nech akceptačný výpočet A na w má k krokov. Potom A_w akceptuje tie slová z L , ktoré A_L akceptuje na najviac k krokov. Takýchto slov je ale len konečne veľa.

- Ak neplatí 3, tak $L_S \notin \mathcal{L}_{RE}$. Nepriamo. Nech $L_S \in \mathcal{L}_{RE}$, nech A_S je TS, ktorý ho akceptuje. Ukážeme, že potom platí 3. Zostrojíme generujúci TS, ktorý bude generovať kódy všetkých konečných jazykov, ktoré majú vlastnosť S .

Lahko zostrojíme TS, ktorý bude generovať kódy všetkých konečných jazykov. Ku každému z nich lahko zostrojíme kód TS, ktorý ho akceptuje. Na každom z týchto kódov TS potrebujeme odsimulovať A_S a vypísat tie kódy konečných jazykov, ktorých TS A_S akceptuje. Keďže ale A_S sa môže aj zacykliť, musíme ho simulovať „naraz na všetkých TS“, nie postupne. Toto sa dá dosiahnuť napr. nasledovne: Striedavo vygenerujeme jeden nový kód konečného jazyka (a zodpovedajúci TS) a odsimulujeme po jednom kroku A_S na každom z už vygenerovaných TS. Vždy, keď A_S niektorý TS akceptuje, kód príslušného konečného jazyka vypíšeme.

- Ukážeme, že ak platí 1, 2, aj 3, tak vieme zostrojiť TS A_S taký, že $L(A_S) = L_S$. A_S na vstupe dostane $\langle A \rangle$ a má zistiť, či $L(A)$ má vlastnosť S . Ak $L(A)$ má túto vlastnosť, tak ju (podľa 2) má aj nejaká jeho konečná podmnožina. Inými slovami, ak žiadna z nich vlastnosť S nemá, tak ani $L(A)$ ju nemá. Budeme skúmať všetky konečné podmnožiny $L(A)$. Ak žiadna z nich vlastnosť S nemá, $\langle A \rangle$ neakceptujeme.¹² Na druhej strane, akonáhle nájdeme nejakú konečnú podmnožinu $L(A)$ s vlastnosťou S , podľa 1 aj $L(A)$ má vlastnosť S – teda A_S akceptuje.

Podľa 3 vieme generovať všetky konečné jazyky s vlastnosťou S . Na všetkých slovách každého z týchto jazykov potrebujeme simulovať A . Použijeme podobný postup ako v predchádzajúcim bode – striedavo vygenerujeme jeden nový konečný jazyk a odsimulujeme jeden krok A na každom zo slov už vygenerovaných jazykov.¹³ Akonáhle A akceptuje všetky slová niektorého konečného jazyka, akceptujeme.

Dôsledok 6.11.8 Nasledujúce vlastnosti rekurzívne vyčísliteľných jazykov nie sú ani čiastočne rozhodnutelné:

- prázdnosť (porušený bod 1, nadmnožiny prázdného jazyka nie sú prázdne)
- rovnosť Σ^* (porušený bod 2)
- $L(A) \in \mathcal{L}_{rec}$ (porušený bod 1)
- $|L(A)| = 1$ (porušený bod 1)
- $L(A) \setminus L_U \neq \emptyset$ (porušený bod 3)

Pristavíme sa pri poslednom tvrdení. Keby sme vedeli vymenovať všetky konečné jazyky s danou vlastnosťou, zjavne vieme vymenovať všetky jednoslovné jazyky s touto vlastnosťou. Jednoslovné jazyky s danou vlastnosťou sú práve jazyky obsahujúce slovo z L_U^C . Upravme stroj, ktorý generuje jednoslovné jazyky s danou vlastnosťou tak, aby namiesto jednoslovných jazykov generoval dotyčné slová. Potom tento stroj generuje L_U^C , čo je spor.

Dôsledok 6.11.9 Nasledujúce vlastnosti rekurzívne vyčísliteľných jazykov sú čiastočne rozhodnutelné:

- neprázdnosť
- $|L(A)| \geq 10$
- $11011 \in L(A)$

9. Kódovanie nečíselných oborov (oborov slov) do prirodzených čísel.

(Úvod)

- konštrukčný objekt - dať na využiť na konečný
 - čas a konečná energie
 - prirodzené čísla $N = \{0, 1, 2, \dots\}$
 - racionálne čísla \mathbb{Q}
 - booleovské funkcie
 - matice } konečné
 - grafy
 - reťazce
- "všetky" sú dajú založovať pomocou prirodzených čísel

konšt. objekt $\rightarrow [kód]$ $\rightarrow \in N$

$m \in N \rightarrow [znamenie] \rightarrow k.$ objekt

- konštruktivnosť N pomocou magistrovítka

1) $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow \dots$

John von Neumann

$$0 = \emptyset = \{\}$$

$$1 = \{\{\}\}$$

$$2 = \{\{\}, \{\{\}\}\}$$

$$3 = \{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}\}$$

:

$$m = \{1, 2, \dots, m-1\}$$

$$m+1 = \{1, 2, \dots, m-1, m\} = m \cup \{m\} \sim 2^m$$

$$\sim(m) = m+1$$

$$2) m = a_k z^k + \dots + a_1 z + a_0$$

$$\text{meno}(m) = a_k \dots a_0 \sim \log m$$

?) Rímske čísla: I, II, III,

(Na prednáške boli kódovanie objektov spojené s Minského strojmi. Kódy v skriptách sa nezhodovali s prednáškou, uprednostnil som kódovanie v skriptách.)

Definícia 6.1. (a) Číslom inštrukcie $(q_i R_j q_m q_n)$ nazveme číslo $p_i^{2c^3(j,m,n)+2}$

(b) Číslom inštrukcie $(q_i R_j P q_k)$ nazveme číslo $p_i^{4c(j,k)+3}$

(c) Číslom inštrukcie $(q_i R_j M q_k)$ nazveme číslo $p_i^{4c(j,k)+1}$

(d) Číslom registrového stroja nazveme súčin čísel všetkých jeho inštrukcií.

Číslo inštrukcie je vždy mocninou prvočísla. Príslušné prvočíslo určuje vnútorný stav, v ktorom sa inštrukcia používa, exponent určuje činnosť pri tomto vnútornom stave. Pretože tento exponent je vždy kladný, a pretože registróv stroj neobsahuje rôzne inštrukcie s rovnakým prvým prvkom, možno číslo M -stroja tvoriť ako súčin čísel jeho inštrukcií. Lahko sa tiež overí nasledujúca veta.

Veta 6.2. Každé kladné celé číslo je číslom práve jedného M -stroja.

Definícia 6.9. (a) Číslom stavu $(q_i; a_0, a_1, \dots, a_n)$ nazveme číslo $2^i \cdot 3^{a_0} \cdot 5^{a_1} \cdots p_{n+1}^{a_n}$.

(b) Číslom konečnej postupnosti stavov (špeciálne, číslom konečného výpočtu) X_0, X_1, \dots, X_n nazveme číslo

$$2^{\text{num}(X_0)} \cdot 3^{\text{num}(X_1)} \cdots p_n^{\text{num}(X_n)}$$

kde $\text{num}(X_i)$ označuje číslo stavu X_i .

$$size(A) = \min\{y | \Sigma_{i=y+1}^A ex(i, A) = 0\} \leq A$$

(Toto sa neprednášalo, ale Šuster mi to v maily odporučil vedieť, najmä def. 9.24)

Definícia 9.1. (a) Číslom T -inštrukcie $(q_i a_j a_k L q_m)$ nazveme číslo $p_{c(i,j)}^{3c(k,m)+1}$

(b) Číslom T -inštrukcie $(q_i a_j a_k N q_m)$ nazveme číslo $p_{c(i,j)}^{3c(k,m)+2}$

(c) Číslom T -inštrukcie $(q_i a_j a_k P q_m)$ nazveme číslo $p_{c(i,j)}^{3c(k,m)+3}$

(d) Číslom Turingovho stroja nazveme súčin čísel všetkých jeho inštrukcií.

Definícia 9.5. (a) Číslom T -stavu $a_{i_0} a_{i_1} \dots a_{i_{k-1}} q_j a_{i_{k+1}} \dots a_{i_n}$, $i_0 \neq 0$, $i_n \neq 0$ budeme nazývať číslo

$$2^{2i_0} \cdot 3^{2i_1} \cdots p_{k-1}^{2i_{k-1}} \cdot p_k^{2j+1} \cdot p_{k+1}^{2i_{k+1}} \cdots p_n^{2i_n}$$

(b) Číslom konečnej postupnosti T -stavov (špeciálne číslom výpočtu) (X_0, X_1, \dots, X_n) budeme nazývať číslo

$$2^{n(X_0)} \cdot 3^{n(X_1)} \cdots p_n^{n(X_n)}$$

kde $n(X_i)$ znamená číslo stavu X_i .

Príklad 9.6. Číslo stavu $OIOq_2IOI$ je $2^2 \cdot 3^4 \cdot 5^2 \cdot 7^5 \cdot 11^4 \cdot 13^2 \cdot 17^4$. Číslo stavu q_1BBIOI je $2^3 \cdot 7^4 \cdot 11^2 \cdot 13^4$. Číslo stavu $IOIBBq_3$ je $2^4 \cdot 3^2 \cdot 5^4 \cdot 13^7$.

Definícia 9.24. (a) Číslom slova $w = a_{i_0} a_{i_1} \dots a_{i_n}$ nazveme číslo

$$\text{num}(w) = 2^{2i_0} \cdot 3^{2i_1} \cdots p_n^{2i_n} \quad (9.24.1)$$

(b) Číselnou množinou priradenou S -množine X budeme nazývať množinu

$$\text{Num}(X) = \{\text{num}(w) \mid w \in X\} \quad (9.24.2)$$

všetkých čísel prvkov množiny X .