Vyčíslitelné funkce

Základy teorie rekurzivních funkcí

Budeme se snažit identifikovat takové funkce, které jsou „spočitatelné“, tj. vyčíslitelné v obecném smyslu (bez ohledu na konkrétní výpočetní systém). Abychom snížili extrémní velikost třídy těchto funkcí, která je dána také varietou definičních oborů a oborů hodnot, omezíme se, uvažujíce možnost kódování, na funkce tvaru:

$$\boxed{f : \mathbb{N}^m \to \mathbb{N}^n}$$

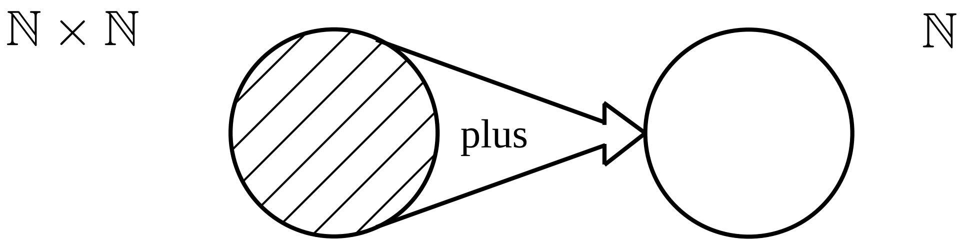
kde $\mathbb{N} = \{0, 1, 2, \ldots\}, \quad m, n \in \mathbb{N}$

❖ Konvence: n-tici $(x_1, x_2, \ldots, x_n) \in \mathbb{N}^n$ budeme označovat jako $\overline{x}$

❖ Klasifikace parciálních funkcí:

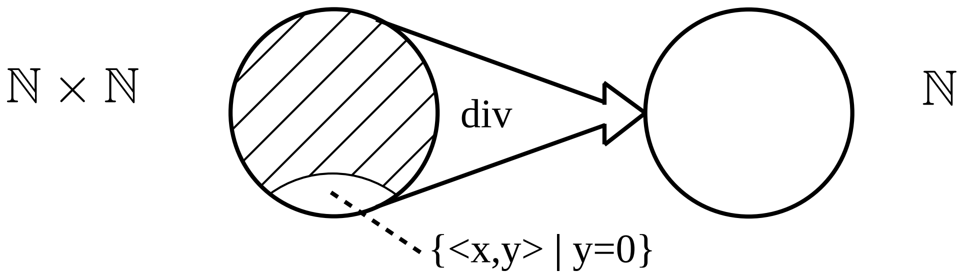
- *Totální funkce*
- *Striktně parciální funkce*

Příklad 11.1 Totální funkce *plus*



$$plus : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$
$$plus(x, y) = x + y$$

Příklad 11.2 Striktně parciální funkce *div*



$$div : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$
$$div(x, y) = \text{celá část } x/y, \text{ je-li } y \neq 0$$

Počáteční funkce

Hiearchie vyčíslitelných funkcí je založena na dostatečně elementárních tzv. *počátečních funkcích*, které tvoří „stavební kameny“ vyšších funkcí.

- Jsou to tyto funkce:

1. *Nulová funkce* (zero function): $\xi() = 0$
   zobrazuje „prázdnou n-tici“ $\mapsto 0$

2. *Funkce následníka* (successor function): $\sigma : \mathbb{N} \to \mathbb{N}$
   $\sigma(x) = x + 1$

3. *Projekce* (projection): $\pi_k^n : \mathbb{N}^n \to \mathbb{N}$
   Vybírá z n-tice k-tou složku, např.: $\pi_2^3(7, 6, 4) = 6$ a $\pi_1^2(5, 17) = 5$
   Speciální případ: $\pi_0^n : \mathbb{N}^n \to \mathbb{N}^0$, tj. např. $\pi_0^3(1, 2, 3) = ()$

Primitivně rekurzivní funkce

Nyní definujme tři způsoby vytváření nových, složitějších funkcí:

1. *Kombinace*:
   Kombinací dvou funkcí $f : \mathbb{N}^k \to \mathbb{N}^m$ a $g : \mathbb{N}^k \to \mathbb{N}^n$ získáme funkci, pro kterou:

$$\boxed{\begin{array}{l} f \times g : \mathbb{N}^k \to \mathbb{N}^{m+n} \\ f \times g(\overline{x}) = (f(\overline{x}), g(\overline{x})),\ \overline{x} \in \mathbb{N}^k \end{array}}$$

   Např.: $\pi_1^3 \times \pi_3^3(4, 12, 8) = (4, 8)$

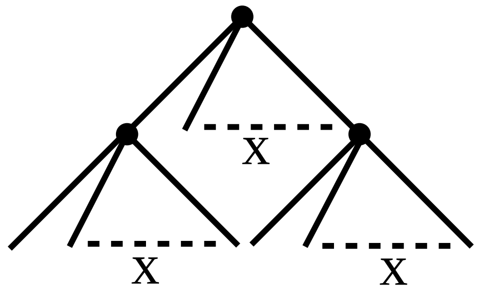
2. *Kompozice*:
   Kompozice dvou funkcí $f : \mathbb{N}^k \to \mathbb{N}^m$ a $g : \mathbb{N}^m \to \mathbb{N}^n$ je funkce, pro kterou:

$$\boxed{\begin{array}{l} g \circ f : \mathbb{N}^k \to \mathbb{N}^n \\ g \circ f(\overline{x}) = g(f(\overline{x})),\ \overline{x} \in \mathbb{N}^k \end{array}}$$

   Např.:
   $\sigma \circ \xi() = 1$
   $\sigma \circ \sigma \circ \xi() = 2$

3. *Primitivní rekurze*:

Příklad 11.3 Předpokládejme, že chceme definovat funkci $f : \mathbb{N}^2 \rightarrow \mathbb{N}$, jejíž hodnoty $f(x, y)$ udávají počet vrcholů x-árního stromu hloubky y, který je pravidelný (úplný):



Zřejmě:

(a) $f(x, 0) = 1$

(b) strom hloubky y má x^y listů. Zvětšením hloubky o jedničku na $y + 1$ musíme přidat $x^{y+1} = x^y \cdot x$ vrcholů

Takže funkci f můžeme definovat následujícím předpisem:

$f(x, 0) = x^0$

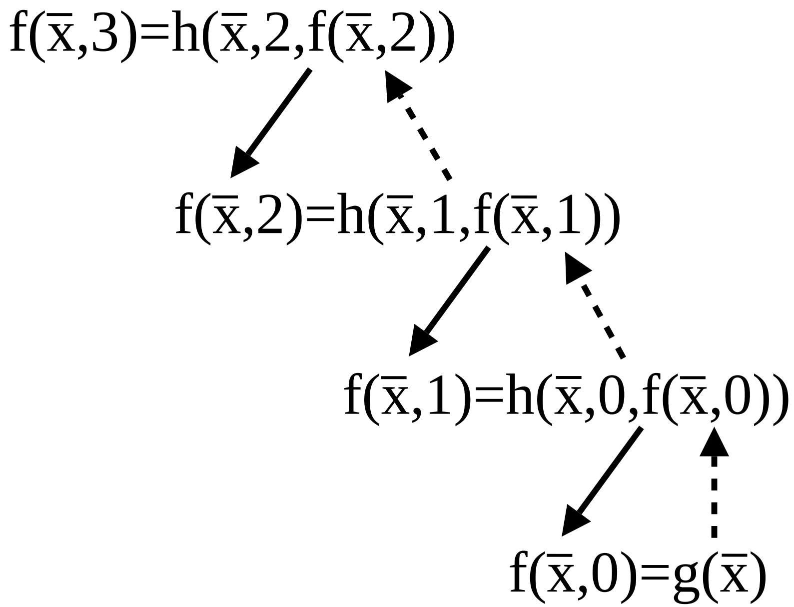
$f(x, y + 1) = f(x, y) + x^y \cdot x$

Např.: $f(3, 2) = f(3, 1) + 3^1 \cdot 3 = f(3, 0) + 3^0 \cdot 3 + 3^1 \cdot 3 = 1 + 3 + 9 = 13$

Primitivní rekurze je technika, která umožňuje vytvořit funkci $f : \mathbb{N}^{k+1} \rightarrow \mathbb{N}^m$ na základě jiných dvou funkcí $g : \mathbb{N}^k \rightarrow \mathbb{N}^m$ a $h : \mathbb{N}^{k+m+1} \rightarrow \mathbb{N}^m$ rovnicemi:

$$\boxed{\begin{array}{l} f(\overline{x}, 0) = g(\overline{x}) \\ f(\overline{x}, y+1) = h(\overline{x}, y, f(\overline{x}, y)), \overline{x} \in \mathbb{N}^k \end{array}}$$

Ilustrace schématu vyčíslení (pro $y = 3$):



Příklad 11.4 Uvažujme funkci $plus : \mathbb{N}^2 \to \mathbb{N}$. Může být definována pomocí primitivní rekurze takto:

$$plus(x, 0) = \pi_1^1(x)$$
$$plus(x, y + 1) = \sigma \circ \pi_3^3(x, y, plus(x, y))$$

což vyjadřuje:

1. $x + 0 = x$
2. $x + (y + 1) = (x + y) + 1 = \sigma(x + y)$

Definice 11.1 *Třída primitivních rekurzivních funkcí* obsahuje všechny funkce, které mohou být z počátečních funkcí vytvořeny:

(a) kombinací

(b) kompozicí

(c) primitivní rekurzí

Věta 11.1 Každá primitivní rekurzivní funkce je totální funkcí.

Důkaz. Počáteční funkce jsou totální. Aplikací kombinace, kompozice a primitivní rekurze na totální funkce dostaneme totální funkce. □

Příklady primitivně rekurzivních funkcí

Třída primitivně rekurzivních funkcí zahrnuje většinu funkcí typických v aplikacích počítačů.

❖ Konvence: Namísto funkcionálních zápisů typu $h \equiv plus \circ (\pi_1^3 \times \pi_3^3)$ budeme někdy používat zápis $h(x, y, z) = plus(x, z)$ nebo $h(x, y, z) = x + z$

❖ *Konstantní funkce*: Zavedeme funkci κ_m^n, která libovolné n-tici $\overline{x} \in \mathbb{N}^n$ přiřadí konstantní hodnotu $m \in \mathbb{N}$

$$\kappa_m^0 \equiv \underbrace{\sigma \circ \sigma \circ \ldots \circ \sigma}_{m-\text{krát}} \xi$$

Také pro $n > 0$ je κ_m^n funkce rekurzivně primitivní:

$$\kappa_m^n(\overline{x}, 0) = \kappa_m^{n-1}(\overline{x})$$
$$\kappa_m^n(\overline{x}, y + 1) = \pi_{n+1}^{n+1}(\overline{x}, y, \kappa_m^n(\overline{x}, y))$$

Např.: $\kappa_3^2(1, 1) = \pi_3^3(1, 0, \kappa_3^2(1, 0)) = \kappa_3^2(1, 0) = \kappa_3^1(1) = \kappa_3^1(0) = \kappa_3^0() = 3$
Kombinací funkcí κ_m^n dostáváme konstanty z $\mathbb{N}^n$, $n > 1$
Např.: $\kappa_2^3 \times \kappa_5^3(x, y, z) = (2, 5)$

❖ *Funkce násobení* :

$$mult(x, 0) = \kappa_0^1(x)$$
$$mult(x, y + 1) = plus(x, mult(x, y))$$

❖ *Funkce umocňování* : $\exp : \mathbb{N}^2 \rightarrow \mathbb{N}$ - analogicky - viz. cvičení

❖ *Funkce předchůdce*:

$$pred(0) = \xi()$$
$$pred(y + 1) = \pi_1^2(y, pred(y))$$

Poznámka: $pred$ je totální funkcí: $pred(0) = 0$

❖ *Funkce monus*:

$$monus(x, 0) = \pi_1^1(x)$$
$$monus(x, y + 1) = pred(monus(x, y))$$

Význam: $monus(x, y) = \begin{cases} x - y & \text{je-li } x \geq y \\ 0 & \text{jinak} \end{cases}$

Notace: $monus(x, y) \equiv x \dot{-} y$

❖ *Funkce eq* (equal): $eg(x,y) = \begin{cases} 1 & \text{je-li } x = y \\ 0 & \text{je-li } x \neq y \end{cases}$

Definice 11.2 $eq(x,y) = 1 \dot{-} ((y \dot{-} x) + (x \dot{-} y))$ nebo formálněji
$eq \equiv monus \circ (\kappa_1^2 \times (plus \circ ((monus \circ (\pi_2^2 \times \pi_1^2)) \times monus \circ (\pi_1^2 \times \pi_2^2))))$

Příklad 11.5 $eq(5,3) = 1 \dot{-} ((3 \dot{-} 5) + (5 \dot{-} 3)) = 1 \dot{-} (0 + 2) = 1 \dot{-} 2 = 0$

❖ *Funkce $\neg$ eq*: $\neg eq \equiv monus \circ (\kappa_1^2 \times eq)$ $\qquad (\equiv 1 \dot{-} eq)$

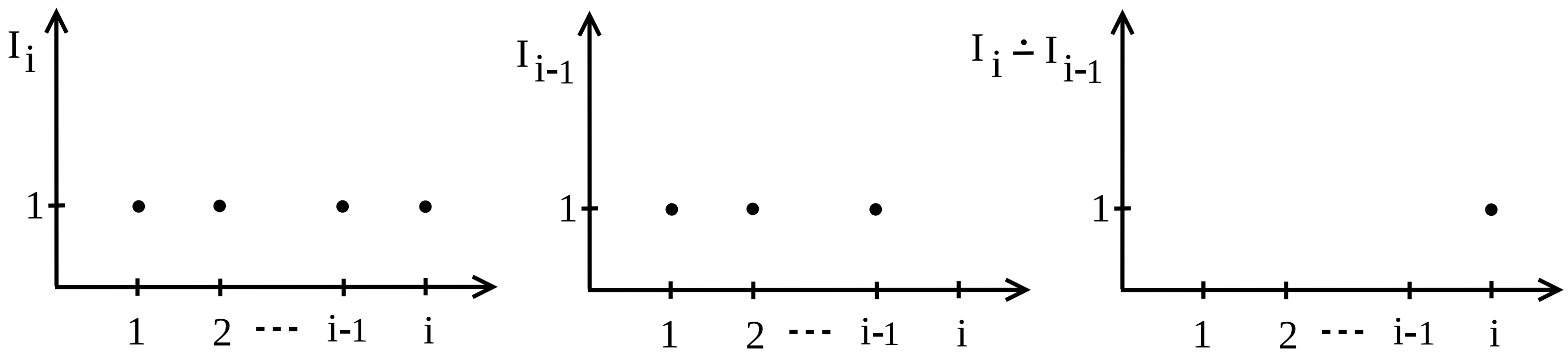
❖ *Tabulární funkce*:

Uvažujme funkce typu:
$$f(x) = \begin{cases} 3 & \text{je-li } x = 0 \\ 5 & \text{je-li } x = 4 \\ 2 & \text{v ostatních případech} \end{cases}$$

které bývají zadávány tabulkou. Tyto funkce lze tvořit pomocí charakteristické funkce

$$\varphi_i(x) = \begin{cases} 1 & \text{je-li } x = i \\ 0 & \text{jinak} \end{cases}$$

která může být vyjádřena jako $monus(I_i, I_{i-1})$, kde $I_i(x) = eq(x \dot{-} i, 0)$



Tabulární funkce mohou být nyní tvořeny konečným součtem násobků konstant a funkcí φ_i a $\neg\varphi_i$.

Např.: nahoře uvedená funkce $f(x)$ je vyjádřitelná ve tvaru:

$$f \equiv mult(3, \varphi_0) + mult(5, \varphi_4) + mult(2, mult(\neg\varphi_0, \neg\varphi_4))$$

❖ *Funkce quo* (quotient): $quo(x,y) = \begin{cases} \text{celá část podílu } x/y \text{ je-li } y \neq 0 \\ 0 \text{ je-li } y = 0 \end{cases}$

Tato funkce může být definována primitivní rekurzí:

$$quo(0,y) = 0$$
$$quo(x+1,y) = quo(x,y) + eq(x+1, mult(quo(x,y),y) + y)$$

Funkce mimo primitivně rekurzivní funkce

Existují funkce, které jsou vyčíslitelné a nejsou primitivně rekurzivními funkcemi. Jsou to všechny striktně parciální funkce (jako div), ale i totální funkce. Taková totální funkce byla prezentována W. Ackermannem (1928) a nazývá se *Ackermannova funkce*. Je dána rovnicemi:

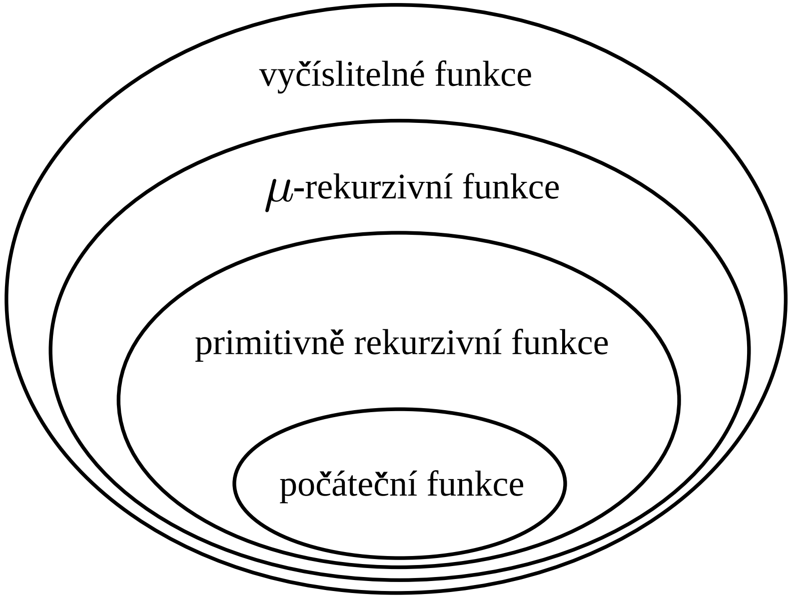
$$\boxed{\begin{aligned} &A(0,y) = y+1 \\ &A(x+1,0) = A(x,1) \\ &A(x+1,y+1) = A(x, A(x+1,y)) \end{aligned}}$$

Věta 11.2 Existuje totální funkce z $\mathbb{N}$ do $\mathbb{N}$, která není primitivně rekurzivní.

Důkaz.
Definice funkcí, které jsou primitivně rekurzivní budeme chápat jako řetězce a můžeme je uspořádat v lexikografickém pořadí s označením $f_1, f_2, \ldots, f_n, \ldots$
Definujeme nyní funkci $f : \mathbb{N} \to \mathbb{N}$ tak, že $f(n) = f_n(n) + 1$ pro $\forall n \in \mathbb{N} \setminus \{0\}$. f je jasně totální a vyčíslitelná. f však není primitivně rekurzivní (kdyby byla, pak $f \equiv f_m$ pro nějaké $m \in \mathbb{N}$. Pak ale $f(m) = f_m(m)$ a ne $f_m(m) + 1$, jak vyžaduje definice funkce f).

□

Definice 11.3 Třída totálních vyčíslitelných funkcí se nazývá *μ-rekurzivní funkce*.



Parciálně rekurzivní funkce

K rozšíření třídy vyčíslitelných funkcí za totální vyčíslitelné funkce zavedeme techniku známou pod názvem *minimalizace*. Tato technika umožňuje vytvořit funkci $f : \mathbb{N}^n \to \mathbb{N}$ z jiné funkce $g : \mathbb{N}^{n+1} \to \mathbb{N}$ předpisem, v němž $f(\overline{x})$ je nejmenší y takové, že:

1. $g(\overline{x}, y) = 0$
2. $g(\overline{x}, z)$ je definována pro $\forall z < y$, $z \in \mathbb{N}$

Tuto konstrukci zapisujeme notací:

$$\boxed{f(\overline{x}) = \mu y[g(\overline{x}, y) = 0]}$$

Příklad 11.6 $f(x) = \mu y[plus(x, y) = 0]$ tj. $f(x) = \begin{cases} 0 \text{ pro } x = 0 \\ \text{nedef. jinak} \end{cases}$

Příklad 11.7 $div(x, y) = \mu t[((x + 1) \dot{-} (mult(t, y) + y)) = 0]$

Příklad 11.8 $i(x) = \mu y[monus(x, y) = 0]$ tj. identická funkce

❖ Funkce definovaná minimalizací je skutečně vyčíslitelná. Výpočet hodnoty $f(\overline{x})$ zahrnuje výpočet $g(\overline{x}, 0), g(\overline{x}, 1), \ldots$ tak douho, pokud nedostaneme:

(a) $g(\overline{x}, y) = 0$ ($f(\overline{x}) = y$)

(b) $g(\overline{x}, z)$ je nedefinována ($f(\overline{x})$ je nedefinována)

Definice 11.4 *Třída parciálně rekurzivních funkcí* je třída parciálních funkcí, které mohou být vytvořeny z počátečních funkcí aplikací:

(a) kombinace

(b) kompozice

(c) primitivní rekurze

(d) minimalizace

Vztah vyčíslitelných funkcí a Turingových strojů

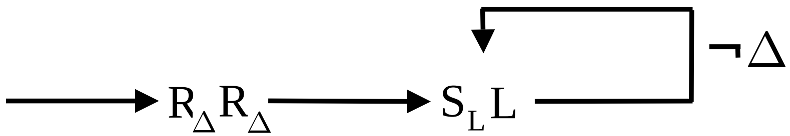
Turingovsky vyčíslitelné funkce

Definice 11.5 Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, q_F)$ *vyčísluje (počítá)* parciální funkci $f : \Sigma^{*m} \to \Sigma_1^{*n}$, $\Sigma_1 \subseteq \Gamma, \Delta \notin \Sigma_1$, jestliže pro každé $(w_1, w_2, \ldots, w_m) \in \Sigma^{*m}$ a odpovídající počáteční konfiguraci $\underline{\Delta} w_1 \Delta w_2 \Delta \ldots \Delta w_m \Delta\Delta\Delta$ stroj M:

1. v případě, že $f(w_1, \ldots, w_m)$ je definována, pak M zastaví a páska obsahuje $\underline{\Delta} v_1 \Delta v_2 \Delta \ldots \Delta v_n \Delta\Delta\Delta$, kde $(v_1, v_2, \ldots, v_n) = f(w_1, \ldots, w_m)$
2. v případě, že $f(w_1, \ldots, w_m)$ není definována, M cykluje (nikdy nezastaví) nebo zastaví abnormálně.

Parciální funkce, kterou může počítat nějaký Turingův stroj se nazývá funkcí *Turingovsky vyčíslitelnou*.

Příklad 11.9



Turingův stroj, který počítá funkci $f(w_1, w_2, w_3) = (w_1, w_3)$.

Příklad 11.10

Nechť L je libovolný jazyk.

Funkce $f(w) = \begin{cases} |w| & \text{jestliže } w \in L \\ 0 & \text{jestliže } w \notin L \end{cases}$ není Turingovsky vyčíslitelná.

Poznámka 11.1 Definice výpočtu funkce Turingovým strojem nepředpokládala speciální pozici hlavy v koncové konfiguraci. Můžeme předpokládat, že M končí v konfiguraci $\underline{\Delta} v_1 \Delta \ldots \Delta v_n \Delta\Delta\Delta$ bez újmy na obecnosti.

Turingovská vyčíslitelnost parciálně rekurzivních funkcí

Budeme uvažovat Turingův stroj s abecedou $\Sigma = \{0, 1\}$ a parciální funkce tvaru $f : \mathbb{N}^m \to \mathbb{N}^n$, m-tice a n-tice budeme kódovat podle vzoru:

$$\boxed{\Delta 11 \Delta 10 \Delta 100 \Delta\Delta \quad \text{reprezentuje trojici} \quad (3, 2, 4)}$$

Věta 11.3 Každá parciálně rekurzivní funkce je Turingovsky vyčíslitelná.

Důkaz.

1. Nejprve je třeba nalézt Turingovy stroje, které vyčíslují počáteční funkce ξ, σ, π.
   (a) $\xi : \to R0L$ (b) (c) viz. cvičení
2. Dále popíšeme konstrukci Turingových strojů pro aplikaci kombinace, kompozice, primitivní rekurze a minimalizace:
   (a) Kombinace:
   Nechť Turingův stroj M_1, resp. M_2 vyčísluje parciální funkci g_1, resp. g_2. Stroj M, který vyčísluje $g_1 \times g_2$ bude 3-páskový Turingův stroj, který začne duplikováním vstupu na 2. a 3. pásku. Pak M simuluje M_1 s využitím pásky 2 a M_2 s použitím pásky 3. Skončí-li M_1 i M_2 řádně, M vyčistí pásku 1 a okopíruje na ní obsah pásek 2 a 3 (oddělených blankem) a zastaví.

Důkaz pokračuje dále.

Pokračování důkazu.

(b) Kompozice:
Funkci $g_1 \circ g_2$ realizuje stroj $\rightarrow M_2 M_1$

(c) Primitivní rekurze:
Uvažujme:

$$f(\overline{x}, 0) = g(\overline{x})$$
$$f(\overline{x}, y+1) = h(\overline{x}, y, f(\overline{x}, y))$$

kde g je parciální funkce, kterou vyčísluje stroj M_1 a h parciální funkce, kterou vyčísluje stroj M_2. Funkci f vyčísluje Turingův stroj, který pracuje takto:

I. Je-li poslední složka vstupu 0, pak vymaže tuto složku, vrátí hlavu na začátek a simuluje stroj M_1

II. Není-li poslední složkou vstupu 0, pak páska musí obsahovat sekvenci $\Delta\overline{x}\Delta y+1\Delta\Delta\Delta$ pro nějaké $y+1>0$
Potom:
(α) S využitím strojů pro kopírování a dekrement transformuj obsah pásky na sekvenci $\Delta\overline{x}\Delta y\Delta\overline{x}\Delta y-1\Delta\ldots\Delta\overline{x}\Delta 0\Delta\overline{x}\Delta\Delta$
Pak přesuň hlavu bezprostředně za 0 a simuluj stroj M_1.

Důkaz pokračuje dále.

Pokračování důkazu.

(β) Nyní bude páska obsahovat
$\Delta\overline{x}\Delta y\Delta\overline{x}\Delta y-1\Delta\ldots\Delta\overline{x}\Delta 0\Delta g(\overline{x})\Delta\Delta$ což je ekvivalentní s
$\Delta\overline{x}\Delta y\Delta\overline{x}\Delta y-1\Delta\ldots\Delta\overline{x}\Delta 0\Delta f(\overline{x},0)\Delta\Delta$
Přesuň hlavu před poslední $\overline{x}$ a simuluj stroj M_2. To vede k
$\Delta\overline{x}\Delta y\Delta\overline{x}\Delta y-1\Delta\ldots\Delta\overline{x}\Delta 1\Delta h(\overline{x},0,f(\overline{x},0))\Delta\Delta$ což je ekvivalentní s
$\Delta\overline{x}\Delta y\Delta\overline{x}\Delta y-1\Delta\ldots\Delta\overline{x}\Delta 1\Delta f(\overline{x},1)\Delta\Delta$

(γ) Pokračuj v aplikaci stroje M_2 na zbytek pásky až je M_2 aplikován na $\Delta\overline{x}\Delta y\Delta f(\overline{x},y)$ a páska je zredukována do tvaru $\Delta h(\overline{x},y,f(\overline{x},y))\Delta\Delta$, což je ekvivalentní žádanému výstupu $\Delta f(\overline{x},y+1)\Delta\Delta$

Důkaz pokračuje dále.

Pokračování důkazu.

(d) Minimalizace:
Uvažujme funkci $\mu y[g(\overline{x}, y) = 0]$, kde parciální funkci g vyčísluje stroj M_1.
Zkonstruujeme 3-páskový stroj M, který pracuje takto:

(1) Zapíše 0 na pásku 2

(2) Zkopíruje obsah pásky 1 a následně pásky 2 na pásku 3

(3) Simuluje stroj M_1 na pásce 3

(4) Jestliže páska 3 obsahuje 0, vymaže pásku 1, okopíruje pásku 2 na pásku 1 a zastaví. Jinak inkrementuje obsah na pásce 2, vymaže pásku 3 a pokračuje krokem (2)

□

Reprezentace Turingového stroje parciálně rekurzivními funkcemi

Abychom dokázali, že Turingova a Churchova teze jsou ekvivalentní, zbývá ukázat, že výpočetní síla Turingových strojů je omezena na možnost vyčíslovat parciálně rekurzivní funkce. K tomu účelu uvažujme Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, q_F)$ a položme $b = |\Gamma|$. Nyní budeme obsah pásky interpretovat jako v opačném pořadí zapsané nezáporné celé číslo při základu b.

Například: Bude-li $\Gamma = \{x, y, \Delta\}$ a interpretujeme-li $x \approx 1, y \approx 2, \Delta \approx 0$, pak páska obsahující $\Delta yx\Delta\Delta y\Delta\Delta\ldots$ je ekvivalentní s $02100200\ldots$ což po reverzi reprezentuje číslo $\ldots 00200120$ při základu 3 a tedy 501.

S touto interpretací můžeme činnost každého Turingova stroje chápat jako výpočet funkce $f : \mathbb{N} \to \mathbb{N}$, která číslu odpovídajícímu počátečnímu obsahu pásky „přiřazuje“ číslo odpovídající obsahu pásky po zastavení stroje. Charakter funkce f specifikuje následující věta.

Věta 11.4 Každý výpočetní proces prováděný Turingovým strojem je procesem vyčíslení nějaké parciálně rekurzivní funkce.

Důkaz.

Budeme vycházet z právě zavedené interpretace činnosti Turingova stroje M, tj. pro každé $n \in \mathbb{N}$, $f(n)$ je definována obsahem pásky při zastavení stroje M. Dále provedeme zakódování stavů: $1 \approx q_0, 0 \approx q_F$, ostatní stavy čísly $2, 3, \dots, k-1$ za předpokladu $|Q| = k$. Tedy jak stavy, tak symboly mají přiřazené numerické hodnoty a můžeme definovat funkce, které sumarizují diagram přechodů stroje M' (M' je zakódovaný stroj M):

$$mov(p, x) = \begin{cases} 2 & \text{jestliže } \delta(p, x) = (*, R) \\ 1 & \text{jestliže } \delta(p, x) = (*, L) \\ 0 & \text{jinak} \end{cases}$$

$$sym(p, x) = \begin{cases} y & \text{jestliže } \delta(p, x) = (*, y) \\ x & \text{jinak} \end{cases}$$

$$state(p, x) = \begin{cases} q & \text{jestliže } \delta(p, x) = (q, *) \\ k & \text{jestliže } p = 0 \text{ nebo } \delta(p, x) \text{ není definována} \end{cases}$$

Funkce sym, mov a $state$ jsou tabulární funkce (totální) a jsou tedy primitivně rekurzivní.

Důkaz pokračuje dále.

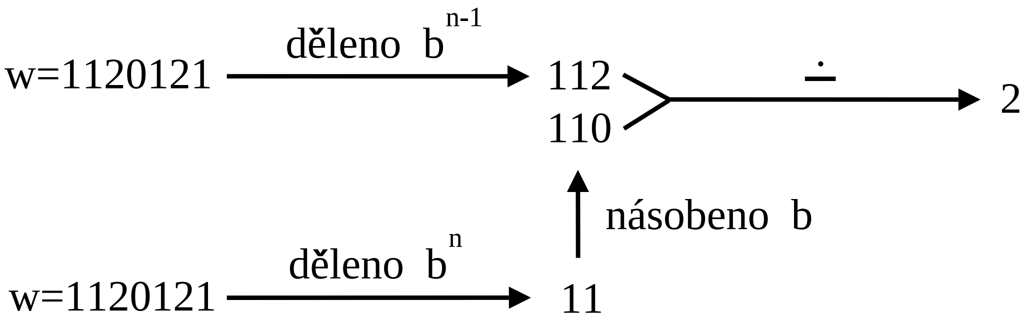
Pokračování důkazu.

Nyní uvažujme konfiguraci Turingova stroje M' ve tvaru trojice (w, p, n), kde w je obsah pásky, p přítomný stav, n pozice hlavy ($n \geq 1$, nejlevější pozice je rovna 1).

Z informace obsažené v konfiguraci může být vypočítán symbol, který se nachází pod hlavou, primitivně rekurzivní funkcí $cursym$:

$$\boxed{cursym(w, p, n) = quo(w, b^{n-1}) \dot{-} mult(b, quo(w, b^n))}$$

Příklad 11.11 Nalezení n-té číslice řetězce $w = 1120121$ pro $n = 5$ a $b = 3$:



$$112 = quo(w, b^{n-1})$$
$$110 = mult(b, quo(w, b^n))$$

Důkaz pokračuje dále.

Pokračování důkazu.

Další funkce, které definujeme na množině konfigurací jsou:

$$\boxed{nexthead(w,p,n) = n \dot{-} eq(mov(p, cursym(w,p,n)), 1) + eq(mov(p, cursym(w,p,n)), 2)}$$

určující příští pozici hlavy (0 indikuje abnormální posuv z 1. pozice vlevo)

$$\boxed{nextstate(w,p,n) = state(p, cursym(w,p,n)) + mult(k, \neg nexthead(w,p,n))}$$

(normálně je 2. sčítanec roven 0; při abnormálním posuvu tato funkce vyčíslí nelegální stav větší než $k-1$)
a konečně funkce

$$\boxed{nexttape(w,p,n) = (w \dot{-} mult(b^{n-1}, cursym(w,p,n))) + mult(b^{n-1}, sym(p, cursym(w,p,n)))}$$

která vyčísluje celé číslo reprezentující nový obsah pásky, po provedení přechodu z konfigurace (w,p,n).
Kombinací tří předchozích funkcí dostaneme funkci $step$, která modeluje 1. krok Turingova stroje, tj. přechod do nové konfigurace:

$$\boxed{step = nexttape \times nextstate \times nexthead}$$

Důkaz pokračuje dále.

Pokračování důkazu.

Nyní definujme funkci $run : \mathbb{N}^4 \to \mathbb{N}^3$; $run(w,p,n,t)$ realizující t přechodů z konfigurace (w,p,n). Opět užijeme primitivní rekurze:

$$\begin{aligned} run(w,p,n,0) &= (w,p,n) \\ run(w,p,n,t+1) &= step(run(w,p,n,t)) \end{aligned}$$

Výsledná funkce vyčíslovaná strojem M' (při vstupu w) je hodnota pásky po dosažení koncového stavu (stavu 0). Počet požadovaných kroků stroje M je dán funkcí $stoptime$

$$stoptime(w) = \mu t[\pi^3_2(run(w,1,1,t)) = 0]$$

Na závěr tedy, je-li $f : \mathbb{N} \to \mathbb{N}$ parciální funkce počítaná strojem M, pak

$$\boxed{f(w) = \pi^3_1(run(w,1,1,stoptime(w)))}$$

z její konstrukce plyne, že f je parciálně rekurzivní funkce. $\square$

Shrňme v obrázku získané informace: