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

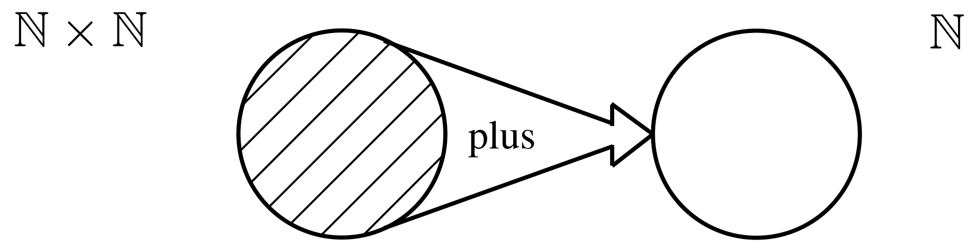
❖ Konvence: n-tici $(x_1, x_2, \ldots, x_n) \in \mathbb{N}^n$ budeme označovat jako $\overline{x}$

❖ Klasifikace parciálních funkcí:

- *Totální funkce*
- *Striktně parciální funkce*

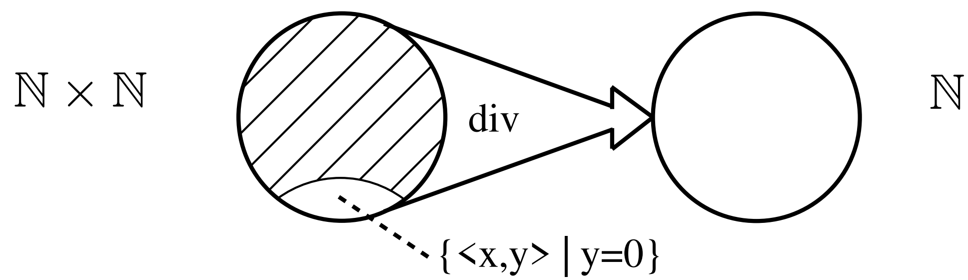
Příklad 11.1 Totální funkce *plus*

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



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

Hierarchie 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 $f : \mathbb{N}^k \rightarrow \mathbb{N}^m$ a $g : \mathbb{N}^k \rightarrow \mathbb{N}^n$ získáme funkci, pro kterou:

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

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

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

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

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

3. *Primitivní rekurze*:

Příklad 11.3 Předpokládejme, že chceme definovat funkci násobení $mult : \mathbb{N}^2 \to \mathbb{N}$.

$$mult(x, y) = \underbrace{x + \cdots + x}_{y}$$

Zřejmě:

(a) Pro $y = 0$ platí $x * 0 = 0$

(b) Pro $y > 0$ je výsledek $x + mult(x, y - 1)$

Takže funkci $mult$ můžeme definovat následujícím předpisem:

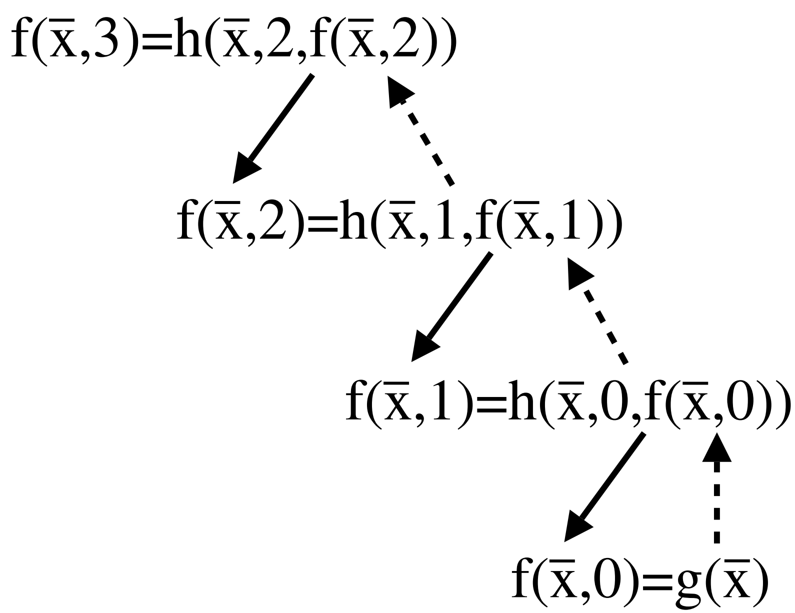
$mult(x, 0) = 0$

$mult(x, y + 1) = x + mult(x, y)$

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

$$\begin{aligned} plus(x,0) &= \pi_1^1(x) \\ plus(x,y+1) &= \sigma \circ \pi_3^3(x,y,plus(x,y)) \end{aligned}$$

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. $\square$

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 $m \in \mathbb{N}$

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

Pro pro $n > 0$ je κ_m^n definována následovně:

$$\kappa_m^n = \kappa_m^0 \circ \pi_0^n$$

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*: $\exp : \mathbb{N}^2 \to \mathbb{N}$ - 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*: $\neg eq \equiv monus \circ (\kappa_1^2 \times eq)$ $(\equiv 1 \dot{-} eq)$

❖ *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:

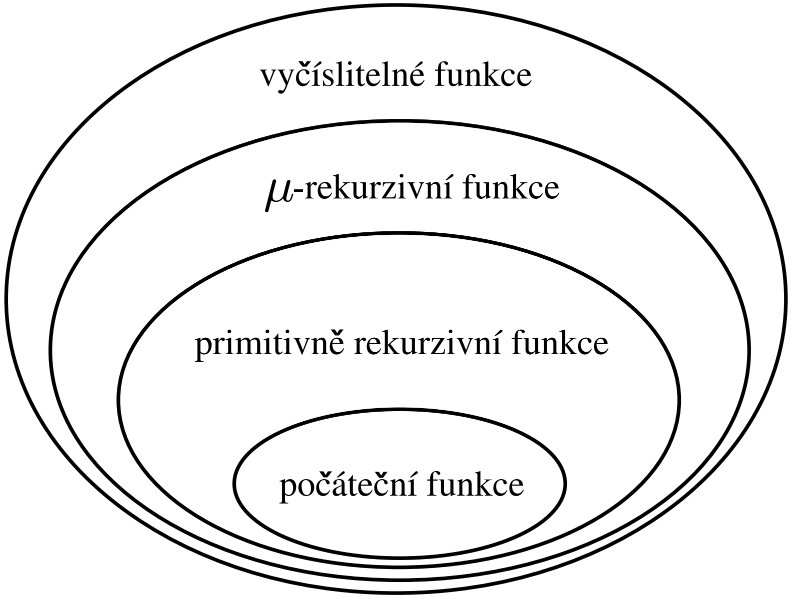
$$\boxed{\begin{aligned} &A(0, y) = y + 1 \\ &A(x + 1, 0) = A(x, 1) \\ &A(x + 1, y + 1) = A(x, A(x + 1, y)) \end{aligned}}$$

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).

$\square$

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 $f : \mathbb{N}^n \rightarrow \mathbb{N}$ z jiné funkce $g : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ 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 dlouho, 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 $f : \Sigma^{*m} \rightarrow \Sigma_1^{*n}$, $\Sigma_1 \subseteq \Gamma, \Delta \notin \Sigma_1$, jestliže pro každé $(w_1, w_2, \ldots, w_m) \in \Sigma^{*m}$ a odpovídající počáteční konfiguraci $\underline{\Delta} w_1 \Delta w_2 \Delta \ldots \Delta w_m \Delta^\omega$ 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 cyklí nebo zastaví abnormálně.

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

Příklad 11.9
Funkci $f(w) = w^R$ je Turingovsky vyčíslitelná.

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.

Turingovská vyčíslitelnost a parciálně rekurzivní funkce

Věta 11.3 Každá parciálně rekurzivní funkce je Turingovsky vyčíslitelná.

Idea důkazu: počáteční funkce, kombinaci, kompozici, projekci, primitivní rekurzi i minimalizaci lze implementovat pomocí Turingova stroje.

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

Idea důkazu: viz další slide.

Idea důkazu

Konfiguraci Turingova stroje lze zakódovat například jako čtveřici následujícím způsobem:

- Pro páskovou abecedu definujme funkci $c : \Gamma \rightarrow < 0, |\Gamma|)$, $c(\Delta) = 0)$.
- *Konfiguraci TS* $\Delta a_1 \dots a_{n-1} a_n \underline{a} b_1 b_2 \dots b_m \Delta^\omega$ ve stavu q_i zakódujeme jako *čtveřici čísel* (v soustavě o základu Γ): $(c(a_1) \dots c(a_n), c(a), c(b_m) \dots c(b_2) c(b_1), i)$
- *Provedení přechodu* $\delta(q_i, a) = (q_j, X)$ na konfiguraci (l, c, r, i) vypočteme následovně:
  - $X = L : (l\ quo\ |\Gamma|, l\ mod\ |\Gamma|, plus(mult(r, |\Gamma|), c), j)$
  - $X = R : (plus(mult(l, |\Gamma|), c), r\ mod\ |\Gamma|, r\ quo\ |\Gamma|, j)$
  - $X = a : (l, c(a), r, j)$
- plus, mult, quo a mod jsou *primitivne rekursivní*.
- Pro přechodovou funkci δ je možné definovat primitivně rekursivní funkci $step_\delta : \mathbb{N}^4 \rightarrow \mathbb{N}^4$ provádějící *jeden krok výpočtu* TS. Pro koncovou konfiguraci x je $step_\delta(x) = x$.
- Dále je možné definovat funkci $final : \mathbb{N}^4 \rightarrow \{0, 1\}$, která vrací 1 pro nekoncovou konfiguraci a 0 pro koncovou.

Idea důkazu

- Nyní definujme následující primitivně rekursivní funkci provádějící k kroků TS:

$$step_k(a, b, c, d, 0) = (a, b, c, d)$$

$$step_k(a, b, c, d, k + 1) = step_\delta(step_k(a, b, c, d, k))$$

- Výpočet počtu kroků nutných k zastavení TS je pak definován jako:

$$nsteps(a, b, c, d) = \mu k[final(steps_k(a, b, c, d, k)) = 0]$$

- Konfigurace v čase zastavení je pak definována pak:

$$result(a, b, c, d) = step_k(a, b, c, d, nsteps(a, b, c, d))$$

Shrňme v obrázku získané informace: