λ-vyčíslitelnost

LKFP přednáška 4, jaro 2024/25

Michal Krupka

8. března 2025

Ukážeme, že λ-definovatelné funkce jsou právě rekurzivní funkce (Church, Turing, Kleene).

1 Rekurzivní funkce

(Množinu $\mathbb{N}$ uvažujeme i s nulou.)

Jsou to *číselné funkce* $\mathbb{N}^m \to \mathbb{N}$ (totální), jejichž hodnoty lze počítat mechanickým postupem.

Iniciální rekurzivní funkce. Jsou všechny totální a jsou tří typů:

1. *Konstanta.* Je to funkce $Z\colon \mathbb{N} \to \mathbb{N}$, $Z(n) = 0$.
2. *Funkce následníka.* $S^+\colon \mathbb{N} \to \mathbb{N}$, $S^+(n) = n + 1$.
3. *itá projekce.* $\pi_i^m\ \mathbb{N}^m \to \mathbb{N}$, $\pi_i^m(\vec{n}) = n_i$.

Totální rekurzivní funkce. Množina rekurzivních funkcí $\mathcal{R}$ je nejmenší množina číselných funkcí obsahující iniciální rekurzivní funkce a dále splňující následující podmínky uzavřenosti na kompozici, primitivní rekurzi a minimalizaci.

Uzavřenost na kompozici. Jestliže funkce $\chi, \psi_1, \ldots, \psi_m \in \mathcal{R}$ (vhodného počtu argumentů) pak funkce φ definovaná předpisem

$$\varphi(\vec{u}) = \chi(\psi_1(\vec{u}), \ldots, \psi_m(\vec{u})) \tag{1}$$

leží v $\mathcal{R}$.

Uzavřenost na primitivní rekurzi. Jestliže funkce ψ o m argumentech a funkce χ o $m+2$ argumentech leží v $\mathcal{R}$, pak i funkce φ o $m+1$ argumentech splňující

$$\begin{aligned}\varphi(0,\vec{n}) &= \psi(\vec{n}),\\ \varphi(k+1,\vec{n}) &= \chi(k,\varphi(k,\vec{n}),\vec{n}) \quad (k \geq 0)\end{aligned} \tag{2}$$

leží v $\mathcal{R}$.

Uzavřenost na minimalizaci. Jestliže funkce $m+1$ argumentů χ je v $\mathcal{R}$, pak i funkce φ, kde $\varphi(\vec{n})$ je nejmenší k takové, že $\chi(\vec{n},k)=0$, je v $\mathcal{R}$.

Značení:

$$\varphi(\vec{n}) = \mu k[\chi(\vec{n},k)=0]. \tag{3}$$

Aby funkce φ byla totální, musí pro každé $\vec{n}$ existovat k tak že $\chi(\vec{n},k)=0$.

Cvičení 1. Ukažte, že následující funkce jsou rekurzivní: 1. konstanta 1, 2. sčítání (pomocí primitivní rekurze), 3. odečítání — zkuste jednak jako totální funkci s tím, že záporný rozdíl nahradíte nulou, a jednak jako parciální funkci, pomocí minimalizace s porušenou podmínkou totálnosti.

2 λ-definovatelnost rekurzivních funkcí

Věta 1 (Kleene)**.** *Totální numerická funkce je rekurzivní právě když je λ-definovalelná.*

Nástin důkazu. Nejprve ukážeme λ-definovatelnost libovolné totální rekurzivní funkce. Začneme tím, že ukážeme λ-definovatelnost iniciálních rekurzivních funkcí.

To bude snadné: Konstantu Z lze definovat termem $\lambda x.\ulcorner 0 \urcorner = \mathrm{K}\ulcorner 0 \urcorner$. Funkce následníka S^+ je definovatelná výrazem S^+ (stejné značení pro dvě různé věci). Projekce π^m_k je definovatelná výrazem

$$\lambda x_0 \dots x_m . x_k.$$

Teď kompozice. Předpokládejme, že funkce $\chi, \psi_1, \dots, \psi_m$ jsou definovány termy (pořadě) $S, T_1, \dots, T_m$. Pak funkci φ z (1) lze definovat termem

$$\lambda \vec{x}.S\left(T_1(\vec{x})\right)\dots\left(T_m(\vec{x})\right)$$

Primitivní rekurze: nechť numerické funkce φ, ψ a χ splňují (2) a funkce ψ a χ jsou definovány termy (pořadě) S a T. Pak φ je definováno termem

$$\mathrm{Y}\lambda f x \vec{y}.(\mathrm{Zero}\, x)(S\vec{x})\big(T(f(P^- x)\vec{y})(P^- x)\vec{y}\big)$$

(ověřte).

A konečně minimalizace. Nejprve předpokládejme, že P je takový výraz, že pro libovolné n se $P\ulcorner n \urcorner$ redukuje na True nebo False. Pak pro

$$H_P \equiv \Theta\big(\lambda hz.(Pz)z(h(S^+z))\big)$$

se výraz

$$\mu P \equiv H_P \ulcorner 0 \urcorner$$

redukuje na nejmenší numerál $\ulcorner n \urcorner$, pro který $P\ulcorner n \urcorner = \mathrm{True}$. Nyní, jestliže funkce χ je λ-definována výrazem S, pak funkce φ z (3) může být definována výrazem

$$\lambda \vec{x}.\mu[\lambda y.\mathrm{Zero}(S\vec{x}y)].$$

Ve druhé části důkazu stručně ukážeme, že je-li totální numerická funkce λ-definovatelná, pak je rekurzivní. Nejprve si uvědomme jednu věc: jelikož standardní numerály jsou (vzájemně různé) normální formy, z Churchova-Rosserova teorému plyne, že rovnost $\ulcorner m \urcorner = \ulcorner n \urcorner$ platí, jedině když $m = n$.

Dále už se jen odkážeme na poznatky o rekurzivních funkcích nesouvisející s λ-kalkulem. Jestliže funkce φ je definována výrazem F, pak její hodnotu v $k$tici $\vec{n}$ získáme převodem aplikace $F\ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$ na normální tvar a interpretací výsledku jako numerálu nějakého čísla. To je algoritmický (rekurzivní) postup, z čehož plyne, že funkce φ je rekurzivní.

To je konec důkazu.

3 Obecné systémy numerálů

Pozoruhodný poznatek: z vlastností standardních numerálů jsme v důkazu použili jen fakt, že máme výrazy na testovaní nulovosti, následníka a předchůdce. Můžeme tedy tvrzení podstatně zobecnit na libovolný *systém numerálů* $\mathbf{d}$, který sestává z uzavřených výrazů $d_0, d_1, \dots$ a výrazů Zero_d a S_d^+ tak, že pro všechna n

$$\begin{aligned} S_d^+ d_n &= d_{n+1}, \\ \mathrm{Zero}\, d_0 &= \mathrm{True}, \\ \mathrm{Zero}\, d_{n+1} &= \mathrm{False}. \end{aligned}$$

Výraz pro předchůdce v $\mathbf{d}$ je výraz P_d^- takový, že pro každé n je

$$P_d^- d_{n+1} = d_n.$$

Řekneme, že numerická funkce φ k argumentů je λ-*definovatelná v* $\mathbf{d}$, jestliže existuje výraz F tak, že pro všechna $\vec{n}$ délky k je

$$Fd_{n_1} \dots d_{n_k} = d_{\varphi(\vec{n})}.$$

Nyní,

Věta 2. *Každá rekurzivní funkce je λ-definovatelná v* $\mathbf{d}$*, právě když v* $\mathbf{d}$ *existuje výraz pro předchůdce.*

Důkaz. Jednoduchým zobecněním důkazu Věty 1.

Pokud například existuje výraz pro předchůdce pro Churchovy numerály (cvičení), platí, že každá rekurzivní funkce je λ-definovatelná pomocí Churchových numerálů.

4 Poznámka parciálním funkcím

Definici rekurzivních funkcí snadno rozšíříme na parciální funkce tím, že u minimalizace připustíme libovolnou funkci χ (je dokázáno, že ale stále stačí, aby byla totální). Narazíme ale na významné problémy.

Za prvé, je třeba jinak definovat term λ-definující kompozici λ-definovatelných funkcí. Například konstanta $\psi = Z$ je, jak víme, λ-definovatelná výrazem $F = \mathrm{K}\ulcorner 0 \urcorner$ a parciální funkce $\varphi \colon \mathbb{N} \to \mathbb{N}$ s prázdným definičním oborem výrazem $G = \mathrm{K}\Omega$. Kompozice $\psi \circ \varphi$ má také prázdný definiční obor, ovšem výraz $\lambda x.F(Gx)$ ji nedefinuje (ověřte).

A za druhé, je třeba rozhodnout, jaké termy mají reprezentovat neexistující hodnotu. Pokud totiž např. parciální funkce φ není definována v čísle n a výraz F funkci λ-definuje, pak aplikace $F\ulcorner n \urcorner$ nesmí mít za normální formu numerál.

To je problém běžně řešený ve funkcionálních jazycích, kde neexistující hodnotu obvykle reprezentuje výraz *Bottom* značený $\perp$.

Způsob, jak jsme v důkazu Věty 1 vyřešili minimalizaci, ukazuje, že výraz pro neexistující hodnotu by vůbec neměl mít normální formu. Dále požadujeme, aby neexistující hodnota byla jen jedna. To nás vede k prvnímu návrhu: za neexistující hodnotu považovat libovolný výraz, který nemá normální formu a k teorii λ přidat axiomy

$$\text{Jestliže } M \text{ a } N \text{ nemají normální formu, pak } M = N.$$

Tato cesta ovšem není správná, protože výsledná teorie by byla nekonzistentní: pro $M \equiv \lambda x.x\mathrm{K}\Omega$ a $N \equiv \lambda x.x\mathrm{S}\Omega$ z $M = N$ odvodíme $M\mathrm{K} = N\mathrm{K}$, což vede k $\mathrm{K} = \mathrm{S}$, což je ovšem, jak víme, kontradikce.

Druhé řešení spočívá v použití výrazů, které nemají *hlavovou normální formu* (hnf). Hlavové normální formy jsou výrazy

$$\lambda x_1 \ldots x_m . y M_1 \ldots M_n.$$

Čísla m a n mohou být i nulová; je-li $m = 0$, jde o aplikaci $y M_1 \ldots M_n$. Proměnná y může být jednou z proměnných $x_1 \ldots x_m$.

Každá normální forma je i hlavovou normální formou. Naopak to neplatí a také neplatí, že by se dvě různé hnf nemohly rovnat.

Výrazy vytvářené v λ-definici minimalizace nemají hlavovou normální formu a takové už lze k reprezentaci neexistujících hodnot použít, když přidáme axiomy

Jestliže M a N nemají hlavovou normální formu, pak $M = N$.

V jazycích s líným vyhodnocováním se používají *slabé hlavové normální formy*, což jsou výrazy tvaru

$$\begin{array}{l} \lambda x.M, \\ x M_1 \ldots M_n \end{array}$$

(tj. tělo funkce a argumenty aplikace se nevyhodnocují).