Fundamentele limbajelor de programare
CURS 9

Traian Florin Șerbănuță și Andrei Sipoș

Facultatea de Matematică și Informatică, DL Info, Anul II
Semestrul II, 2023/2024

Feluri de limbaje funcționale

Până acum am studiat două modele matematice ale programării funcționale:

- λ-calculul fără tipuri
  - un model de calcul complet, având și parțialitate
  - un limbaj de programare modelat oarecum pe baza lui: Lisp
- λ-calcul cu tipuri (simple sau mai complicate)
  - modelează o subclasă strictă a clasei funcțiilor calculabile totale
  - un limbaj de programare oarecum în acest stil: Agda

Pentru a modela limbajul Haskell, avem nevoie de un model de calcul complet care să includă și un sistem de tipuri. În continuare, vom prezenta un exemplu de așa ceva, anume PCF (Programming Computable Functions), introdus de Scott și Plotkin în anii '70.

Tipuri și termeni

Tipurile vor fi următoarele:

- un singur tip de bază, $\mathbb{N}$ (reprezentând numerele naturale);
- pentru orice tipuri ρ, τ, avem un tip $\rho \to \tau$ (reprezentând funcțiile parțiale „de la ρ la τ").

Vom defini termenii fără tipuri asociate, și apoi le vom aloca tipuri „à la Curry":

- avem variabile, λ-abstracțiuni și aplicații exact ca la λ-calculul fără tipuri;
- z este termen („zero");
- dacă M este termen, atunci $s(M)$ este termen („succesor");
- dacă M, N, P sunt termeni, iar w este variabilă, atunci $\mathit{ifz}(M, N, w, P)$ este termen;
- dacă M este termen, iar w este variabilă, atunci $\mathit{fix}(w, M)$ este termen.

Alocarea tipurilor

Acestea sunt regulile de deducție pentru alocarea de tipuri termenilor:

$$\frac{}{\Gamma \cup \{x : \sigma\} \vdash x : \sigma}$$

$$\frac{\Gamma \cup \{x : \sigma\} \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

$$\frac{}{\Gamma \vdash z : \mathbb{N}} \qquad \frac{\Gamma \vdash M : \mathbb{N}}{\Gamma \vdash s(M) : \mathbb{N}}$$

$$\frac{\Gamma \vdash M : \mathbb{N} \quad \Gamma \vdash N : \tau \quad \Gamma \cup \{w : \mathbb{N}\} \vdash P : \tau}{\Gamma \vdash \mathit{ifz}(M, N, w, P) : \tau} \qquad \frac{\Gamma \cup \{w : \tau\} \vdash M : \tau}{\Gamma \vdash \mathit{fix}(w, M) : \tau}$$

Valori

Următoarele reguli stabilesc „când ceva este valoare”:

$$\frac{}{val(z)} \qquad \frac{val(M)}{val(s(M))} \qquad \frac{}{val(\lambda x.M)}$$

Reguli de evaluare

Următoarele reguli stabilesc evaluarea termenilor în regim *eager*:

$$\frac{M \to M'}{s(M) \to s(M')} \qquad \frac{M \to M'}{ifz(M, N, w, P) \to ifz(M', N, w, P)}$$

$$\frac{}{ifz(z, N, w, P) \to N} \qquad \frac{val(s(M))}{ifz(s(M), N, w, P) \to P[w := M]}$$

$$\frac{M \to M'}{MN \to M'N} \qquad \frac{val(M) \quad N \to N'}{MN \to MN'}$$

$$\frac{val(N)}{(\lambda x.M)N \to M[x := N]} \qquad \frac{}{fix(w, M) \to M[w := fix(w, M)]}$$

Exemple de termeni

Operația de adunare este reprezentată de următorul termen PCF (de ce?):

$$fix(f, \lambda x.\lambda y.ifz(y, x, w, s((fx)w))).$$

Putem defini operația de „scădere cu punct", pentru orice x, $y \in \mathbb{N}$, prin:

$$x \dot{-} y = \begin{cases} x - y, & \text{dacă } x \geqslant y, \\ 0, & \text{altfel.} \end{cases}$$

Ea se poate reprezenta prin (de ce?):

$$fix(f, \lambda x.\lambda y.ifz(y, x, w, ifz(x, z, u, (fu)w))).$$

Notăm acest termen tot cu $\dot{-}$.

Exemple de termeni

Un termen care reprezintă funcția de cel mai mare divizor comun, obținut prin algoritmul lui Euclid, este:

$$fix(f, \lambda x.\lambda y.ifz(x, y, w, ifz(y, x, u, ifz((\dot{-}y)x, f((\dot{-}x)y)y, v,$$

$$(f(x))(((\dot{-}y)x))))))).$$

În sfârșit, definim operatorul de *minimizare nemărginită*, μ, pentru orice funcție parțială f, punând $\mu(f)$ ca fiind, în caz că există (altfel, rămâne nedefinit), cel mai mic număr natural n cu proprietatea că $f(n) = 0$ și că f este definită pentru toate numerele mai mici decât n. Un termen care îl reprezintă pe μ este:

$$\lambda f.((fix(g, \lambda n.\lambda f.ifz(fn, n, w, (g(Sn))f))z)f).$$