Fundamentele limbajelor de programare

Programare funcțională. Lambda-calcul cu tipuri simple. Inferența tipurilor. Normalizare.

Traian Florin Șerbănuță și Andrei Sipoș

Facultatea de Matematică și Informatică, DL Info

Anul II, Semestrul II, 2024/2025

Secțiunea 1

Algoritmul de inferență a tipurilor

Reguli de deducție pentru tipuri (a la Curry)

- $\Gamma \uplus \{x : \sigma\} \vdash x : \sigma \qquad (\textsc{var})$
- $\dfrac{\Gamma \cup \{x : \sigma\} \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad (\textsc{abs})$
- $\dfrac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M\ N : \tau} \qquad (\textsc{app})$

Algoritmul de inferență a tipurilor

Pornim cu un λ termen fără tipuri în care toate variabilele legate au fost redenumite cu variabile noi. Atunci putem asocia fiecărei variabile x un tip X, unde X e variabilă.

Algoritm simplificat

$$
\begin{aligned}
c(x, Z) &:= \{X = Z\} \\
c(\lambda x.M, Z) &:= c(M, W) \cup \{Z = X \to W\} \\
c(M\ N, Z) &:= c(M, W_1) \cup c(N, W_2) \cup \{W_1 = W_2 \to Z\}
\end{aligned}
$$

Corectitudine Dacă θ unificator pentru $c(M, Z)$, atunci $\Gamma_\theta \vdash M : \theta(Z)$, unde $\Gamma_\theta = \{x : \theta(X) \mid x \text{ variabilă}\}$.

Completitudine $\Gamma \vdash M : \sigma$ implică există θ unificator pentru $c(M, Z)$ cu
- $\Gamma(x) = \Gamma_\theta(x) = \theta(X)$ pentru orice $x \in FV(M)$ și
- $\theta(Z) = \sigma$

Corectitudinea algoritmului de inferență

Dacă θ unificator pentru $c(M, Z)$, atunci $\Gamma_\theta \vdash M : \theta(Z)$, unde $\Gamma_\theta = \{x : \theta(X) \mid x \text{ variabilă}\}$.

Demonstrație

Inducție după structura lui M

- $M = x$: $c(M, Z) = \{X = Z\}$ evident, deoarece $x : \theta(X) \in \Gamma_\theta$ și $\theta(X) = \theta(Z)$ (regula VAR).
- $M = \lambda x.N$: $c(M, Z) = c(N, W) \cup \{Z = X \to W\}$.
  Fie θ unificator. Atunci unificator pentru $c(N, W)$ și $\theta(Z) = \theta(X) \to \theta(W)$.
  Din ip. inducție, $\Gamma_\theta \vdash N : \theta(W)$ Dar $x : \theta(X) \in \Gamma_\theta$, deci $\Gamma_\theta \vdash \lambda x.N : \theta(X) \to \theta(W) = \theta(Z)$ (regula ABS).
- $M = N\ P$: $c(M, Z) = c(N, W_1) \cup c(P, W_2) \cup \{W_1 = W_2 \to Z\}$
  Fie θ unificator. Atunci unificator pentru $c(N, W_1)$, $c(P, W_2)$ și $\theta(W_1) = \theta(W_2) \to \theta(Z)$.
  Din ip. inducție, $\Gamma_\theta \vdash N : \theta(W_1) = \theta(W_2) \to \theta(Z)$ și $\Gamma_\theta \vdash P : \theta(W_2)$, de unde $\Gamma_\theta \vdash N\ P : \theta(Z)$ (regula APP).

Completitudinea algoritmului de inferență

Dacă $\Gamma \vdash M : \sigma$ atunci există θ unificator pentru $c(M, Z)$ cu

- $\Gamma(x) = \Gamma_\theta(x)$ pentru orice $x \in FV(M)$ și
- $\theta(Z) = \sigma$

Demonstrație (inducție după $\Gamma \vdash M : \sigma$)

- $\Gamma \uplus \{x : \sigma\} \vdash x : \sigma$ (VAR)
  $c(x, Z) = \{X = Z\}$, Aleg $\theta(X) = \theta(Z) = \sigma$
- $\dfrac{\Gamma \cup \{x : \sigma\} \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau}$ (ABS)
  $c(\lambda x.M, Z) = c(M, W) \cup \{Z = X \to W\}$
  Din ip. ind, fie θ unificator pt $c(M, W)$ astfel încât $\theta(W) = \tau$ și $\Gamma(x) = \Gamma_\theta(x)$ pentru orice $x \in FV(M)$. Deoarece Z nu apare în $c(M, W)$, pot alege $\theta' = \theta[Z := \theta(X) \to \theta(W)]$ și observa că satisface condițiile.

Completitudinea algoritmului de inferență

Demonstrație (cont.)

- $$\frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M\ N : \tau}\ (\textsc{app})$$

$c(M\ N, Z) = c(M, W_1) \cup c(N, W_2) \cup \{W_1 = W_2 \to Z\}$

Fie θ_M unificator pt $c(M, W_1)$ astfel încât $\theta_M(W_1) = \sigma \to \tau$ și $\Gamma(x) = \Gamma_{\theta_M}(x)$ pentru orice $x \in FV(M)$.

Fie θ_N unificator pt $c(N, W_2)$ astfel încât $\theta_N(W_2) = \sigma$ și $\Gamma(x) = \Gamma_{\theta_N}(x)$ pentru orice $x \in FV(N)$.

Definesc $\theta(U) := \begin{cases} \theta_M(U), & \text{dacă } U \text{ apare în } c(M, W_1), \\ \theta_N(U), & \text{dacă } U \text{ apare în } c(N, W_2), \\ \tau, & \text{dacă } U = Z \end{cases}$

E bine definit, deoarece variabilele comune între $c(M, W_1)$ și $c(N, W_2)$ pot fi doar variabilele libere, pe care trebuie să fie de acord cu Γ.

Secțiunea 2

Normalizare

Ce este Normalizarea?

: Normalizarea slabă

orice termen bine format se poate (beta-)reduce la o formă normală.

Normalizare puternică

orice termen bine format se (beta-)reduce la o formă normală *pe orice cale*.

Teorema de normalizare

În Calculul Lambda cu Tipuri Simple (CLTS), toți termenii bine formați sunt puternic normalizabili.

Reguli de bună formare (a la Church)

1. $x : A$ *dacă* x are tipul A (Variabilă)
2. $\dfrac{t : B}{\lambda x.\ t : A \to B}$ *dacă* x are tipul A (Abstracție)
3. $\dfrac{t : A \to B \qquad u : A}{t\ u : B}$ (Aplicație)

Substituția și α-echivalența

Substituția și α-echivalența sunt bine definite pentru termeni cu tipuri:

- Substituția $t : A[x := N : B]$ este definită când tipul lui x este B și rezultatul are tipul A
  - se demonstrează (relativ) ușor prin inducție asupra lui $t : A$
- Alpha-echivalența permite înlocuirea variabilelor legate cu variabile *de același tip*

Beta-reducția

- $(\lambda x.\ t)\ u \Rightarrow t[x := u]$ (BETA)
  - se observă că dacă x are tipul A, și $t : B$ atunci și $u : A$, pentru ca să putem avea termenul $(\lambda x.\ t)\ u : B$
  - deci substituția este bine definită și termenul obținut e de același tip ca redex-ul
- Plus regulile de compatibilitate cu abstracția și aplicație (și ele sunt bine definite)

  Reducerile pot apărea oriunde într-un termen (nu doar la vârf).

$$\frac{t \Rightarrow t'}{\lambda x.\ t \Rightarrow \lambda x.\ t'} \qquad \frac{t \Rightarrow t'}{t\ u \Rightarrow t'\ u} \qquad \frac{u \Rightarrow u'}{t\ u \Rightarrow t\ u'}$$

Subject reduction (conservarea tipului)

Lemă

Dacă $t : A$ și $t \Rightarrow t'$ atunci $t' : A$.

Demonstrație

Inducție după relația de rescriere, folosind buna definiție a substituției pentru cazul de bază.

Forme normale (sintactic)

O formă normală este un termen care nu are nici un redex. Poate fi definită prin reguli:

- $FN(x)$ (FNVAR)
- $\dfrac{FN(t)}{FN(\lambda x.t)}$ (FNABS)
- $\dfrac{FN(u)}{FN(x\ u)}$ (FNAPP1)
- $\dfrac{FN(t\ u) \qquad FN(v)}{FN((t\ u)\ v)}$ (FNAPP2)

Strategia generală

1. Definim noțiunea de **SN (strongly normalizing)**.
2. Definim o interpretare a tipurilor prin mulțimi de termeni SN.
3. Demonstrăm că toți termenii bine formați aparțin acestor mulțimi.

Normalizare Puternică (SN)

- Un termen este **puternic normalizabil** dacă toate secvențele de reduceri beta se termină.
  - Adică: nu există reduceri infinite.
- Inductiv: cea mai mică mulțime care conține formele normale și e închisă la regula: dacă pentru orice t' astfel încât $t \to t'$ avem că t' e în mulțime, atunci și t e în mulțime
- $SN(t)$ *dacă* $FN(t)$ (SNFN)
- $$\frac{SN(t') \text{ pentru orice } t' \text{ pentru care } t \Rightarrow t'}{SN(t)}$$ (SNACC)

Interpretarea tipurilor

Fie $[[A]]$ interpretarea semantică a unui tip A, definită recursiv astfel:

Tipuri de bază $[[\alpha]]$ conține exact termeni puternic normalizabili de tip α

$$[[\alpha]] = \{t : \alpha \mid SN(t)\}$$

Tipuri săgeată $[[A \to B]]$ conține acei termeni care aplicați tuturor termenilor din $[[A]]$ sunt în $[[B]]$

$$[[A \to B]] := \{t \mid \forall u \in [[A]],\ t\ u \in [[B]]\}$$

Proprietăți ale interpretării

- Dacă $t \in [[A \to B]]$ și $u \in [[A]]$, atunci $t\ u \in [[B]]$.
- Dacă $t \in [[A]]$ atunci $t : A$
- Toți termenii din interpretare sunt puternic normalizabili.
  - Prin inducție: dacă $SN(t\ u)$ atunci și $SN(t)$

$[[A]]$ conține variabilele de tip A

Lema

Pentru orice variabilă x de tip $A_1 \to \cdots \to A_n \to A$, $n \geq 0$ și orice secvență de termeni $t_i \in [[A_i]]$, $x\ t_1 \cdots t_n \in [[A]]$

Demonstrație: inducție după A.

- $A = \alpha$. Reiese din definiție, deoarece $SN(x\ t_1 \cdots t_n)$ (nu adaugă redexuri noi față de t_i).
- $A = B \to C$. Fie $u \in [[B]]$. Instanțiem ipoteza de inducție pentru C cu $n+1$ și $A_{n+1} = B$ și $t_{n+1} = u$. Deci, $x\ t_1 \cdots t_n\ u \in [[C]]$ Deoarece u ales arbitrar, $x\ t_1 \cdots t_n \in [[B \to C]]$

Corolar

Dacă x e de tip A, atunci $x \in [[A]]$

$[[A]]$ e închisă la reducție

Lema

Dacă $t \in [[A]]$ și $t \Rightarrow t'$ atunci $t' \in [[A]]$.

Demonstrație: inducție după A.

- Dacă $A = \alpha$, atunci $SN(t)$ și deci și $t' : A$ și $SN(t')$
- Dacă $A = B \to C$, fie $u \in [[B]]$, arbitrar. Avem că $t\ u \in [[C]]$. Aplicând ipoteza de inducție pentru C și $t\ u \Rightarrow t'\ u$, reiese că $t'\ u \in [[B]]$
  Deoarece u arbitrar ales, reiese că $t' \in [[B \to C]]$

Substituție compatibilă

Definiție

O substituție de la variabile la termeni se numește compatibilă (cu interpretarea) dacă duce orice variabilă de tip A într-un termen din $[[A]]$, interpretarea lui A.

Proprietăți

- Substituția identitate este compatibilă.
- Dacă σ compatibilă, $t\sigma \in [[A]]$ și $t \Rightarrow t'$, atunci $t'\sigma \in [[A]]$.

Teoremă: Toți termenii cu tipuri simple sunt în SN

Demonstrație

Deoarece substituția identitate este compatibilă și interpretarea unui tip conține doar termeni în SN, este suficient să demonstrăm următorul rezultat: Dacă $t : A$ atunci pentru orice substituție compatibilă σ, $t\sigma \in [[A]]$.

Teoremă: Toți termenii cu tipuri simple sunt în SN

Suficient: Dacă $t : A$ atunci pentru orice substituție compatibilă σ, $t\sigma$ este bine format și aparține lui $[[A]]$.

Demonstrație: inducție pe definiția lui $t : A$

- $x : A$ *dacă* x are tipul A (Variabilă)
  Atunci $t\sigma = \sigma(x) \in [[A]]$ din definiția lui σ
- $\dfrac{t : A \to B \qquad u : A}{t\ u : B}$ (Aplicație)
  Din ipoteza de inducție, $t\sigma \in [[A \to B]]$ și $u\sigma \in [[A]]$, de unde $(t\ u)\sigma = t\sigma\ u\sigma \in [[B]]$

Teoremă: Toți termenii cu tipuri simple sunt în SN

Demonstrație (continuare)

$$\frac{t : B}{\lambda x.\ t : A \to B}\ \textit{dacă } x \text{ are tipul } A \qquad (\textsc{Abstracție})$$

Din ipoteza de inducție, pentru orice σ compatibilă, $t\sigma \in [[B]]$.

Fie σ compatibilă. Vrem ca $(\lambda x.\ t)\sigma \in [[A \to B]]$

E suficient să arătăm următorul rezultat mai general (instanțiat pentru $n = 0$):

Dacă $t\sigma \in [[A_1 \to \cdots A_n \to B]]$ pentru orice σ compatibilă și x de tip A_0 și $u_i \in [[A_i]]$ pentru $0 \leq i \leq n$, atunci pentru orice σ,

$(\lambda x.\ t)\sigma\, u_0\, u_1 \cdots u_n \in [[B]]$.

Teoremă: Toți termenii cu tipuri simple sunt în SN

Suficient: Dacă $t\sigma \in [[A_1 \to \cdots A_n \to B]]$ pentru orice σ compatibilă și x de tip A_0 și $u_i \in [[A_i]]$ pentru $0 \leq i \leq n$, atunci pentru orice σ, $(\lambda x.\ t)\sigma u_0\ u_1 \cdots u_n \in [[B]]$

Demonstrație

Demonstrație prin inducție după B

$B = A_{n+1} \to B'$: trebuie să arăt că pentru orice σ,
$(\lambda x.\ t)\sigma u_0 u_1 \cdots u_n \in [[A_{n+1} \to B']]$.
Fie u_{n+1} arbitrar ales. Aplic ipoteza de inducție pentru B' și $t\sigma \in [[A_1 \to \cdots A_n \to A_{n+1} \to B']]$ și x de tip A_0 și u_i, $0 \leq i \leq n+1$ și obțin $(\lambda x.\ t)\sigma u_0\ u_1 \cdots u_n\ u_{n+1} \in [[B']]$
Concluzia urmează din faptul că u_{n+1} arbitrar ales.

Teoremă: Toți termenii cu tipuri simple sunt în SN

Suficient: Dacă $t\sigma \in [[A_1 \to \cdots A_n \to B]]$ pentru orice σ compatibilă și x de tip A_0 și $u_i \in [[A_i]]$ pentru $0 \leq i \leq n$, atunci pentru orice σ, $((\lambda x.\ t)\sigma)u_0\ u_1 \cdots u_n \in [[B]]$

Demonstrație (continuare)

Dacă $B = \alpha$, trebuie să arăt că $SN(((\lambda x.\ t)\sigma)u_0 u_1 \cdots u_n)$.
Reducere la absurd. Avem $(\lambda x.\ t)\sigma = \lambda x.\ (t\sigma')$ unde $\sigma' = \sigma[x := x]$.
Fie $(\lambda x.\ t\sigma')u_0 u_1 \cdots u_n = t_0 \overset{n_0}{\Longrightarrow} t_1 \overset{n_1}{\Longrightarrow} \cdots$ o secvență infinită.

Fie $t_k \overset{n_k}{\Longrightarrow} t_{k+1}$ primul pas pentru care $n_k = 1$ (trebuie să existe, pentru că toți ceilalți redecși sunt în $t\sigma'$ șau u_i pentru vreun i și ei sunt in SN.
Atunci t_k e de forma $(\lambda x.\ t'\sigma'')u_0' u_1' \cdots u_n'$ unde $t \Rightarrow^* t'$, $u_i \Rightarrow^* u_i'$ și pentru orice $y \in FV(t) \setminus \{x\}$, $\sigma'(y) \Rightarrow^* \sigma''(y)$, iar pentru orice alt y, $\sigma''(y) = \sigma'(y)$. Deci $t_{k+1} = (t'\sigma''[x := u_0'])u_1' \cdots u_n'$.
Din $t\sigma \in [[A_1 \to \cdots A_n \to \alpha]]$ pentru orice σ, iese că $t'\sigma \in [[A_1 \to \cdots A_n \to \alpha]]$ pentru orice σ și aleg $\sigma = \sigma''[x := u_0']$, apoi folosind definiția interpretării iese că $(t'\sigma''[x := u_0'])u_1' \cdots u_n' \in [[\alpha]]$ deci în SN, contradicție.

Consecințe ale normalizării

- Nu există bucle infinite în CLTS.
- Programele scrise în acest sistem se termină întotdeauna.
- CLTS nu este Turing complet.
- Putem demonstra confluența mai simplu
  - Local confluență și terminare implică confluență

Concluzie

- Calculul Lambda cu Tipuri Simple are proprietatea de normalizare puternică.
- Am demonstrat acest lucru prin interpretarea semantică a tipurilor în termeni puternic normalizabili (Metoda lui Tait).
- Sistemul oferă garanția că orice funcție este totală.

Referințe

- H. Barendregt, *The Lambda Calculus: Its Syntax and Semantics*