Poglavje 7

TIPI

Lambda račun je bil razširjen s tipi na dva načina. Haskell Curry je definiral λ-račun s tipi leta 1932, Alonzo Church pa leta 1940. Dve vrsti λ-računa s tipi sta vodila do dveh različnih družin sistemov.

V Curryevi teoriji tipov ostanejo λ-izrazi brez tipov. Vsak izraz predstavlja množico možnih tipov. Ta množica je lahko prazna, vsebuje en element ali neskončno število elementov.

V Churchovi teoriji tipov so vsi λ-izrazi označeni s tipi. Vsak izraz ima (z upoštevanjem ekvivalenčne relacije) natačno en tip, ki se običajno lahhko izpelje iz anotacije tipa izraza.

Curryev in Churchev pristop k lambda računu s tipi ustrezata dvem pristopom k obravnavanju tipov v programskih jezikih. Prvi pristop obravnava programe, ki so lahko tudi ne vsebujejo oznak tipov izrazov. V tem primeru prevajalnik preveri ali je mogoče izpeljati tipe iz programa. To se zgodi samov primeru, da je program pravilen.

Zelo razširjen primer takšnega jezika je ML [6]. Za takšne sisteme pravimo, da imajo *implicitne tipe*.

Drugi pristop k tipom v programskih jezikov je striktno označevanje tipov izrazov. Preverjanje tipov v takšnih jezikih je običajno lažje, ker ni potrebno konstruirati tipov.

Za takšne jezike pravimo, da imajo *eksplicitne tipe*. Programski jeziki, ki uporabljajo Churchev sistem tipov so iz Algolove družine jezikov npr. Pascal kot tudi Java.

Nekateri avtorji opisujejo Curryev λ-račun s tipi kot λ-račun *s prirejanjem tipov*, Churchev sistem pa λ-račun *s tipi*.

7.1 Curryev sistem tipov

Implicitni izračun tipov je predlagal Curry leta 1934 za teorijo kombinatorjev. Kasneje, leta 1958 je teorijo modificiral za λ-račun. S teorijo priredimo elemente dane množice tipov $\mathbb{T}$ izrazom λ-računa. Iz tega razloga Curryev lambda račun s tipi velikokrat imenujemo *sistem za prirejanje tipov*.

Če priredimo tip $\sigma \in \mathbb{T}$ λ-izrazu $M \in \Lambda$ to zapišemo $\vdash M : \sigma$. Običajmo uporabljamo množico predpostavk Γ za izpeljavo tipa izraza in zapišemo $\Gamma \vdash M : \sigma$. Pravimo, da predpostavke Γ vodijo do $M : \sigma$ ali, da $M : \sigma$ izpeljemo iz Γ.

Konkreten sistem za izpeljavo tipov je odvisen od množice $\mathbb{T}$ in od množice pravil za prirejanje tipov.

Definicija 7.1.1. *Množica tipov $\lambda_{\rightarrow}$, ki jo označimo z $\mathbb{T}$, je induktivno definirana na sledeč način.*

$$\begin{array}{ll} \alpha, \alpha', \ldots \in \mathbb{T} & \textit{(spremenljivke tipov)} \\ \sigma, \tau \in \mathbb{T} \Rightarrow (\sigma \rightarrow \tau) \in \mathbb{T} & \textit{(funkcijski tipi)} \end{array}$$

Ker bomo $\mathbb{T}$ velikokrat uporabljali dodajmo še naslednjo definicijo v abstraktni sintaksi.

$$\begin{array}{lcll} \mathbb{T} & = & \mathbb{V} \mid \mathbb{T} \rightarrow \mathbb{T} & \text{(vsi tipi)} \\ \mathbb{V} & = & \alpha \mid \mathbb{V}' & \text{(spremenljivke tipov)} \end{array}$$

Opazka 7.1.1. 1. *Naj bo $\sigma_1, \ldots, \sigma_n \in \mathbb{T}$, potem izraz*

$$\sigma_1 \rightarrow \sigma_2 \rightarrow \ldots \rightarrow \sigma_n$$

pomeni

$$(\sigma_1 \rightarrow (\sigma_2 \rightarrow \ldots (\sigma_{n-1} \rightarrow \sigma_n)..)).$$

Operator $\rightarrow$ je desno asociativen.

2. $\alpha, \beta, \gamma, \ldots$ *predstavljajo spremenljivke tipov.* □

Definicija 7.1.2. 1. Stavek *je izraz oblike $M : \sigma$, kjer je $M \in \Lambda$ in $\sigma \in \mathbb{T}$. Tip σ predstavlja* predikat *in M predstavlja* subjekt *stavka.*

2. Deklaracija *je stavek kjer je subjekt spremenljivka.*

3. Osnova[1] *je množica deklaracij, ki imajo različne spremenljivke kot subjekte.*

[1] V literaturi koncept *osnova* srečamo pod imenom *kontekst*.

Definicija 7.1.3. *Stavek* $M : \sigma$ *je* izpeljiv iz *konteksta* Γ *kar zapišemo:*

$$\Gamma \vdash M : \sigma,$$

če lahko $\Gamma \vdash M : \sigma$ *izpeljemo z naslednjo množico pravil.*

$$\frac{(x : \sigma) \in \Gamma}{\Gamma \vdash x : \sigma} \tag{7.1}$$

$$\frac{\Gamma \vdash M : (\sigma \to \tau) \quad \Gamma \vdash N : \sigma}{\Gamma \vdash (M\,N) : \tau} \tag{7.2}$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash (\lambda x.M) : (\sigma \to \tau)} \tag{7.3}$$

Opazka 7.1.2. 1. *Če želimo poudariti uporabljen sistem tipov napišemo* $\Gamma \vdash_{\lambda_\to} M : \sigma$.

2. *Pravilo 7.2 imenujemo* $\to$*-predstavitev, pravilo 7.3 pa* $\to$*-eliminacija.*

3. *Izraz* $\Gamma, x : \sigma$ *pomeni* $\Gamma \cup \{x : \sigma\}$.

4. *Če* $\Gamma = \{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$ *potem lahko namesto* $\Gamma \vdash M : \sigma$ *napišemo* $x_1 : \sigma_1, \ldots, x_n : \sigma_n \vdash M : \sigma$. □

Naslednja definicja pravil poudari uporabo naravne dedukcije pri izpeljavah. Pri tej definiciji ne uporabljamo osnove Γ.

$$\frac{M : (\sigma \to \tau) \quad N : \sigma}{(M\,N) : \tau} \qquad \frac{\begin{array}{c}\cancel{x : \sigma} \\ \vdots \\ M : \tau\end{array}}{(\lambda x.M) : (\sigma \to \tau)} \tag{7.4}$$

Prečrtana predpostavka $\cancel{x : \sigma}$ pomeni, da smo iz $x : \sigma$ izpeljali $M : \tau$ iz česar smo sklepali na $(\lambda x.M) : (\sigma \to \tau)$. Pri tem ne potrebujemo več predpostavke $x : \sigma$ in jo lahko *izbrišemo*.

Primer 7.1.1. 1. *Uporabimo pravila za izpeljavo tipov za izraz* $\lambda x.\lambda y.x$.

$$\frac{\dfrac{x : \sigma, y : \tau \vdash x : \sigma}{x : \sigma \vdash (\lambda y.x) : (\tau \to \sigma)}}{\vdash (\lambda x.\lambda y.x) : (\sigma \to \tau \to \sigma)}$$

2. *Pokažemo lahko, da za vsak* $\sigma \in \mathbb{T}$ *velja*

$$\vdash (\lambda x.x) : (\sigma \to \sigma).$$

3. *Še en primer z nepraznim osnovo.*

$$y : \sigma \vdash (\lambda x.x)y : \sigma.$$

7.1.1 Lastnosti $\lambda_{\rightarrow}$

Prva lastnost, ki si jo bomo ogledali analizira kakšna *osnova* je potrebna, da bi izpeljati dodelitev tipov.

Definicija 7.1.4. *Naj bo $\Gamma = \{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$ osnova.*

1. *Γ lahko obravnavamo kot parcialno funkcijo. Velja $dom(\Gamma) = \{x_1, \ldots, x_n\}$ in $\Gamma(x_i) = \sigma_i$.*

2. *Naj bo V_0 množica spremenljivk. Potem $\Gamma \restriction V_0 = \{x : \sigma | x \in V_0 \wedge \sigma = \Gamma(x)\}$.*

3. *Substitucija $[\tau/\alpha]\sigma$ zamenja $\tau \in \mathbb{T}$ za $\alpha \in \mathbb{V}$ v $\sigma \in \mathbb{T}$.*

Lema 7.1.1 (Lema osnove $\lambda_{\rightarrow}$)**.** *Naj bo Γ osnova.*

1. *Če je $\Gamma' \supseteq \Gamma$ druga osnova potem $\Gamma \vdash M : \sigma \Rightarrow \Gamma' \vdash M : \sigma$.*

2. *$\Gamma \vdash M : \sigma \Rightarrow FV(M) \subseteq dom(\Gamma)$.*

3. *$\Gamma \vdash M : \sigma \Rightarrow \Gamma \restriction FV(M) \vdash M : \sigma$.*

Dokaz.

1. Z indukcijo na izpeljavo $M : \sigma$.

   Primer 1. $M : \sigma = x : \sigma \in \Gamma \Rightarrow x : \sigma \in \Gamma' \wedge \Gamma' \vdash M : \sigma$.

   Primer 2. $M : \sigma = (M_1\, M_2) : \sigma \Rightarrow \exists\tau[M_1 : (\tau \rightarrow \sigma) \wedge M_2 : \tau]$.
   Po induktivni hipotezi velja $\Gamma' \vdash M_1 : (\tau \rightarrow \sigma)$ in $\Gamma' \vdash M_2 : \sigma$. Velja torej tudi $\Gamma' \vdash (M_1\, M_2) : \sigma$.

   Primer 3. $M : \sigma = (\lambda x : M_1) : (\sigma_1 \rightarrow \sigma_2) \Rightarrow \Gamma, x : \sigma_1 \vdash M_1 : \sigma_2$. Lahko predpostavljamo, da $x \notin dom(\Gamma')$. Potem $\Gamma', x : \sigma_1$ razširi $\Gamma, x : \sigma_1$. Po induktivni hipotezi velja $\Gamma', x : \sigma_1 \vdash M_1 : \sigma_2$ in torej tudi $\Gamma' \vdash (\lambda x.M_1) : (\sigma_1 \rightarrow \sigma_2)$.

2. Z indukcijo na izpeljavo $M : \sigma$. Poglejmo si samo primer $M : \sigma = (\lambda x.M_1) : (\sigma_1 \rightarrow \sigma_2)$. Takoj sledi $\Gamma, x : \sigma_1 \vdash M_1 : \sigma_2$. Naj bo $y \in FV(\lambda x.M_1)$ potem $y \in FV(M_1)$ in $y \not\equiv x$. Po induktivni hipotezi velja $y \in dom(\Gamma, x : \sigma_1)$ potem velja tudi $y \in dom(\Gamma)$

3. Po indukciji na izpeljavo $M : \sigma$. Obravnavali bomo samo primer $M : \sigma = (M_1\, M_2) : \sigma$. Direktno sledi $\exists\tau[M_1 : (\tau \rightarrow \sigma) \wedge M_2 : \tau]$. Po induktivni hipotezi velja $\Gamma \restriction FV(M_1) \vdash M_1 : (\tau \rightarrow \sigma)$ in $\Gamma \restriction FV(M_2) \vdash M_2 : \sigma$. Zaradi (1) sledi $\Gamma \restriction FV(M_1\, M_2) \vdash M_1 : (\tau \rightarrow \sigma)$ in $\Gamma \restriction FV(M_1\, M_2) \vdash M_2 : \sigma$. Velja torej tudi $\Gamma \restriction FV(M_1\, M_2) \vdash (M_1\, M_2) : \sigma$.

$\square$

Druga lastnost $\lambda_{\rightarrow}$, ki nas zanima je kako izrazi določene oblike dobijo tip. Uporabno je tudi, da vemo, da nekateri izrazi nimajo tipa.

Lema 7.1.2 (Tvorna lema $\lambda_{\rightarrow}$)**.** *1.* $\Gamma \vdash x : \sigma \Rightarrow (x : \sigma) \in \Gamma$.

2. $\Gamma \vdash M\, N : \tau \Rightarrow \exists\sigma[\Gamma \vdash M : (\sigma \rightarrow \tau) \wedge \Gamma \vdash N : \sigma]$.

3. $\Gamma \vdash \lambda x.M : \rho \Rightarrow \exists\sigma, \tau[\Gamma, x : \sigma \vdash M : \tau \wedge \rho \equiv (\sigma \rightarrow \tau)]$.

Dokaz. Z indukcijo po dolžini izpeljave. □

Lema 7.1.3 (Tipi podizrazov $\lambda_{\rightarrow}$)**.** *Naj bo M' pod-izraz M. Potem velja $\Gamma \vdash M : \sigma \Rightarrow \Gamma' \vdash M' : \sigma'$ za nek Γ' in σ'.*

Dokaz. Z indukcijo po tvorbi M. □

Zgornja lema pravi, da če ima M tip $\Gamma \vdash M : \sigma$ potem imajo vsi pod-izrazi tudi tip.

Naslednji izrek bomo potrebovali za dokaz *varnosti* λ-računa [15], ki jo sestavlja *napredek* redukcije λ-izrazov in *ohranjanje* tipov pri redukciji.

Izrek 7.1.1 (Napredek redukcije $\lambda_{\rightarrow}$)**.** *Naj bo $M \in \Lambda$ in $\Gamma \vdash M : \sigma$.*

1. možnost: $M : \sigma$ je v normalni obliki.

2. možnost: $\exists M' : M \rightarrow_{\beta} M'$.

Dokaz. Uporabimo indukcijo po strukturi M (ali tudi po izpeljavi tipa M).

Primer 1. $M : \sigma = x : \sigma$ je primerek 1. možnosti.

Primer 2. $M : \sigma = (M_1\, M_2) : \sigma$.

Po indukcijski hipotezi je M_1 bodisi v normalni obliki ali lahko naredimo korak evaluacije; enako velja za M_2. Če lahko M_1 razvijemo za en korak, potem lahko uporabimo pravilo za aplikacijo $M_1 \rightarrow_{\beta} M_1' \Rightarrow (M_1\, M_2) \rightarrow_{\beta} (M_1'\, M_2)$ nad M.

Če je M_1 v normalni obliki in M_2 ni potem lahko uporabimo pravilo $M_2 \rightarrow_{\beta} M_2' \Rightarrow (M_1\, M_2) \rightarrow_{\beta} (M_1\, M_2')$ nad M.

Končno, če je $M_1 = (\lambda x.M_3)$ potem je M redeks in lahko izvedemo pravilo za aplikacijo $(\lambda x.M_3)\, M_2 \rightarrow_{\beta} [M_2/x]M_3$. Sicer, če sta M_1 in M_2 normalni obliki in M_1 ni λ-abstrakcija, potem je M normalna oblika.

Primer 3. $M : \sigma = (\lambda x.M_1) : \sigma$. Podobno kot v primeru (2) lahko predpostavimo, da za M_1 lastnost velja. Če obstaja redukcija $M_1 \rightarrow_{\beta} M_1'$ potem lahko naredimo korak redukcije tudi za M. Če je M_1 v normalni obliki je tudi M normalna oblika.

□

Izrek o *napredku* izpeljave je posledica normalizacijskega izreka λ-računa s tipi, ki pravi, da vse redukcijske strategije vodijo do izraza v normalni obliki. Normalizacijski izrek je predstavljen kasneje.

Naslednja lema je ena izmed ključnih pri dokazu izreka o *ohranitvi* tipov pri redukciji izraza (Izrek 7.1.2).

Lema 7.1.4 (Substitucijska lema $\lambda_{\rightarrow}$)**.** *1.* $\Gamma \vdash M : \sigma \Rightarrow [\tau/\alpha]\Gamma \vdash M : [\tau/\alpha]\sigma$.

2. Naj bo dan $\Gamma, x : \sigma \vdash M : \tau$ *in* $\Gamma \vdash N : \sigma$. *Potem velja tudi* $\Gamma \vdash [N/x]M : \tau$.

Dokaz. 1. Po indukciji na izpeljavo $M : \sigma$.

2. Po indukciji na izpeljavo stavka $\Gamma, x : \sigma \vdash M : \tau$. Za dano izpeljavo $M : \tau$ obdelamo primere na končnem pravilu uporabljenem pri tvorbi $M : \tau$.

   Pravilo 7.1:
   $$M = z, \text{ kjer je } z : \tau \in (\Gamma, x : \sigma)$$

   Imamo dva primera, ki jih je potrebno obdelati: $z = x$ in z je neka druga spremenljivka. V primeru $z = x$ velja $[N/x]z = N$ kot tudi $\tau = \sigma$. Zahtevan rezultat $\Gamma \vdash N : \sigma$ je med predpostavkami! V drugem primeru pa velja $[N/x]z = z$, kjer je rezultat direkten.

   Pravilo 7.2:
   $$\begin{array}{l} M = M_1\ M_2 \\ \Gamma, x : \sigma \vdash M_1 : (\tau_2 \rightarrow \tau_1) \\ \Gamma, x : \sigma \vdash M_2 : \tau_2 \\ \tau = \tau_1 \end{array}$$

   Po indukcijski hipotezi velja $\Gamma \vdash [N/x]M_1 : (\tau_2 \rightarrow \tau_1)$ in $\Gamma \vdash [N/x]M_2 : \tau_2$. Ker velja $\Gamma \vdash (M_1\ M_2) : \tau$ in ker po definiciji evaluacije substitucije na funkcijskem klicu velja $[N/x](M_1\ M_2) = ([N/x]M_1\ [N/x]M_2)$, potem velja $\Gamma \vdash ([N/x]M_1\ [N/x]M_2) : \tau_1 \equiv [N/x](M_1\ M_2) : \tau$.

   Pravilo 7.3:
   $$\begin{array}{l} M = \lambda y.M_1 \\ \tau = \tau_2 \rightarrow \tau_1 \\ \Gamma, x : \sigma, y : \tau_2 \vdash M_1 : \tau_1 \end{array}$$

   Lahko predpostavimo, da velja $x \neq y$ in $y \notin FV(N)$. Z uporabo permutacije na podizpeljavi dobimo $\Gamma, y : \tau_2, x : \sigma \vdash M_1 : \tau_1$. Z uporabo oslabitve na dani izpeljavi $\Gamma \vdash N : \sigma$ dobimo $\Gamma, y : \tau_2 \vdash N : \sigma$. Po indukcijski hipotezi velja $\Gamma, y : \tau_2 \vdash [N/x]M_1 : \tau_1$. Po definiciji evaluacije substitucije velja $[N/x](\lambda y.M_1) = (\lambda y.[N/x]M_1)$. Zaradi indukcijske hipoteze velja $\Gamma \vdash (\lambda y.[N/x]M_1) : (\tau_2 \rightarrow \tau_1)$ in torej $\Gamma \vdash [N/x](\lambda y.M_1) : (\tau_2 \rightarrow \tau_1) = [N/x]M : \tau$.

$\square$

Naslednji izrek pravi, da je množica $M \in \Lambda$, ki imajo določen tip zaprta za redukcijo.

Izrek 7.1.2 (Ohranjanje tipov $\lambda_\rightarrow$)**.** *Naj bo $M \in \Lambda$ in $M \rightarrow_\beta^* M'$.*

$$\Gamma \vdash M : \sigma \Rightarrow \Gamma \vdash M' : \sigma$$

Dokaz. Uporabili bomo indukcijo po konstrukciji relacije $\rightarrow_\beta^*$. Pokažimo najbolj pomemben primer: evaluacijo redeksa.

Naj bo $M : \sigma = (\lambda x.P)\, Q$ in $M' = [Q/x]P$. Če velja $\Gamma \vdash (\lambda x.P)\, Q : \sigma$ potem zaradi tvorne leme 7.1.2 velja, da obstaja τ tako da $\Gamma \vdash (\lambda x.P) : (\tau \rightarrow \sigma)$ oz. $\Gamma, x : \sigma \vdash P : \sigma$ in $\Gamma \vdash Q : \tau$.

Zaradi indukcijske hipoteze velja ohranitev tipov za P in Q. Zaradi substitucijske leme 7.1.4 potem velja $\Gamma \vdash [Q/x]P : \sigma$. □

Zaradi izreka 7.1.1 in izreka 7.1.2 velja *varnost tipov* λ-računa s tipi. Varnost včasih imenujemo tudi *uglašenost* λ-računa[2].

Izrazi iz Λ, ki imajo tip, pa niso zaprti za razširitev. Na primer,

$$\vdash I : (\sigma \rightarrow \sigma), \text{ vendar } \nvdash K\, I\, (\lambda x.x\, x) : (\sigma \rightarrow \sigma).$$

Lastnost velja v splošnem.

Opazka 7.1.3. *Naj bodo $M, M' \in \Lambda$ in $\sigma, \sigma' \in \mathbb{T}$ tako da $M' \rightarrow_\beta^* M$.*

$$\vdash M : \sigma \textit{ in } \vdash M' : \sigma' \textit{ vendar } \nvdash M' : \sigma.$$

Dokaz. Naj bo $M \equiv \lambda x.\lambda y.y$, $M' \equiv S\, K$, $\sigma \equiv \alpha \rightarrow (\beta \rightarrow \beta)$ in $\sigma' \equiv (\beta \rightarrow \alpha) \rightarrow (\beta \rightarrow \beta)$. □

7.2 Normalizacija $\lambda_\rightarrow$ izrazov

Lambda izraz M *strogo normalizira*, če se vse možne sekvence redukcije z začetkom M ustavijo.

Obstajajo izrazi, ki imajo normalno obliko vendar ne normalizirajo strogo, ker imajo neskončno sekvenco redukcije. Na primer $KI\Omega =_\beta I$, čeprav je sekvenca $\Omega \rightarrow_\beta \Omega \rightarrow_\beta \ldots$ neskončna.

[2]Klasično pomeni uglašenost pravil teorije, da vse kar izpeljemo s pravili so korektni stavki. Enako velja tudi za izpeljave λ-izrazov—β-redukcija vodi do izrazov korektnega tipa.

Uglašenost se tipično uporablja skupaj s *kompletnostjo*, ki pravi, da dana pravila teorije izpeljejo vse možne stavke. Kaj natančno predstavlja kompletnost prirejanja tipov λ-izrazov?

Če bi v primeru $KI\Omega$ uporabljali strategijo klic-po-vrednosti se evaluacija nebi nikoli zaključila. Strategija je torej pomembna pri iskanju normalne oblike izraza.

Naslednji izrek zagotavlja izračun normalno obliko z redukcijo skrajno levega redeksa.

Izrek 7.2.1 (Normalizacijski izrek)**.** *Če ima M normalno obliko potem jo dobimo z iteriranjem redukcij skrajno levih redeksov—redeksov, ki imajo λ skrajno levo.*

Dokaz normalizacijskega izreka se nahaja v knjigi H.Barendregta [2]. V nadaljevanju si bomo pogedali še dva drugačna dokaza normalizacije λ-izrazov.

Šibek normalizacijski izrek pravi, da obstaja neka strategija za normalizacijo izraza, ki se zaključi. *Strog normalizacijski izrek* pravi, da se vse možne strategije normalizacije zaključijo.

7.2.1 Šibek normalizacijski izrek

Poglejmo si varianto lambda računa $\lambda_{\rightarrow}$, ki bo uporabljena za prikaz šibke normalizacije $\lambda_{\rightarrow}$.

$$\begin{array}{llcll} Tip & \tau & ::= & \tau_1, \ldots, \tau_n & osnovni\ tipi \\ & & | & \tau_1 \times \tau_2 & produkt \\ & & | & \tau_1 \rightarrow \tau_2 & funkcije \\ Izraz & M & ::= & x_1 : \tau_1, \ldots, x_n : \tau_n & spremenljivke \\ & & | & \langle M, N \rangle & pari \\ & & | & \Pi^1 M & projekcija \\ & & | & \Pi^2 M & projekcija \\ & & | & \lambda x.M & abstrakcija \\ & & | & M\ N & aplikacija \end{array} \tag{7.5}$$

Pri izračunu normalne oblike izraza M imamo na vsakem koraku več možnosti pri izbiri redeksa, ki reduciramo. Možnih izbir je končno mnogo.

Šibki normalizacijski izrek pravi, da lahko najdemo pravilno redukcijo, ki vodi do normalne oblike. Ne izkl učuje pa možnosti slabih redukcij, ki ne vodijo v normalno obliko. Zato govorimo o šibki normalizaciji.

Izrek 7.2.2 (Šibka normalizacija)**.** *Naj bo M izraz lambda računa brez tipov. Obstaja metoda za evaluacijo M, ki se zaključi.*

Dokaz bo sestavljen iz večih delov. Najprej bomo definirali stopnjo izraza s katero bomo lahko spremljali velikost izraza med redukcijo.

Definicija 7.2.1. *1. Stopnja $\rho(\tau)$ tipa τ je definirana kot:*

(a) $\rho(\tau)$ *= 1 če je τ atomičen.*

(b) $\rho(\tau_1 \times \tau_2) = \rho(\tau_1 \rightarrow \tau_2) = max(\rho(\tau_1), \rho(\tau_2)) + 1.$

2. *Stopnja $\rho(r)$ redeksa r je definirana z:*
   (a) $\rho(\Pi^1\langle M,N\rangle) = \rho(\Pi^2\langle M,N\rangle) = \rho(\tau_1 \times \tau_2)$, *kjer je $\tau_1 \times \tau_2$ tip $\langle M,N\rangle$.*
   (b) $\rho((\lambda x.M)\ N) = \rho(\tau_1 \rightarrow \tau_2)$, *kjer je $\tau_1 \rightarrow \tau_2$ tip $(\lambda x.M)$.*
3. *Stopnja $d(M)$ izraza M je maksimum stopenj redeksov, ki jih vsebuje. Izraz M v normalni obliki ima stopnjo $d(M)$ enako 0.* □

Opazka 7.2.1. *1. Atomičen tip τ v (1a) lahko stoji za večimi redukcijami, ki se ovrednotijo v tip τ.*

2. *Stopnja tipa τ definirana z (1b) predstavlja višino drevesa s katerim predstavimo τ, plus ena.*
3. *Stopnja redeksa (2) je definirana na osnovi stopnje tipa redeksa! Stopnja redeksa $(\lambda x.M)\,N$ je definirana kot $\rho(\tau_1 \rightarrow \tau_2)$ kjer je $x : \tau_1$, $N : \tau_1$ in $(\lambda x.M) : \tau_1 \rightarrow \tau_2$.*

   Recimo da bodisi M ali N vsebuje redeks npr. $((\lambda x.M')\,N') : \tau_3$. Stopnjo tega redeksa se v kontekstu zunanjega redeksa izračuna kot $\rho(\tau_3)$. Stopnja vgnezdenega redeksa se torej ne upošteva pri izračunu stopnje zunanjega redeksa.

   Po drugi strani lahko M ali N vsebujeta λ-abstrakcije (brez aplikacije), ki predstavljajo funkcije tipa $\tau_1 \rightarrow \tau_2$, kar zviša stopnjo M ali N.

   Stopnja redeksa $(\lambda x.M)\,N$ torej predstavlja stopnjo tipa $(\lambda x.M) : \tau_1 \rightarrow \tau_2$. S tem upoštevamo največje število možnih redukcij, ki bi lahko nastane zaradi strukture M oz. N. Redukcije ni nujno, da se realizirajo v danem kontekstu.
4. *Stopnja izraza M je maksimalna stopnja vseh redeksov v M. Če izraz nima redeksa ima stopnjo 0.*
5. *Redeks R ima dve stopnje: kot redeks in kot izraz. Ker redeks lahko vsebuje druge redekse velja $d(R) \geq \rho(R)$.* □

Ob uporabi substitucije se lahko zmanjša stopnja izraza. Velikost izraza, ki je rezultat substitucije lahko ocenimo na naslednji način.

Lema 7.2.1. *Če je x tipa τ potem $d([N/x]M) \leq max(d(M), d(N), \rho(\tau))$.*

Dokaz. V izrazu $[N/x]M$ imamo lahko:

1. redekse, ki sestavljajo M (kjer je x postal N),
2. redeksi N, ki zamenjajo vse pojavitve x,
3. nove redekse v primeru, da se x pojavlja v kontekstu $\Pi^1 x$ (ali $\Pi^2 x$ ali $x\ L$) in je $N = \langle N', N''\rangle$ ali $\lambda y.N'$. Te novi redeksi imajo stopnjo τ.

□

Poglejmo najprej lastnost, ki jo bomo potrebovali kasneje.

Lema 7.2.2. *Naj bo $R : \tau$ redeks. Potem velja $\rho(R) > \rho(\tau)$.*

Dokaz. Naj bo $R : \tau = (\lambda x.M)\, N : \tau$ potem obstaja $\sigma : (\lambda x.M) : \sigma \to \tau$. Trivialno velja $\rho(R) = \rho(\sigma \to \tau) > \rho(\tau)$! □

Poglejmo zdaj še razmerje med stopnjo izraza in stopnjo reduciranega izraza.

Lema 7.2.3. *Če $M \to_\beta^* N$ potem $d(N) \leq d(M)$.*

Dokaz. Obravnavati moramo samo primer, ko imamo eno samo redukcijo: $M \to_\beta N$, če zamenjamo R s R'. Stanje je zelo podobno tistemu iz prejšnje Leme. N vsebuje naslednje komponente.

- Redekse, ki so bili v M in ne tudi v R spremenjeni z zamenjavo R s R' (kar ne vpliva na stopnjo).
- Redekse iz R', ki smo jih dobili s poenostavitvijo R ali z interno substitucijo v R: $(\lambda x.L)\, L'$ postane $[L'/x]L$, Lema 7.2.1 pa pravi, da $d(R') \leq max(d(L), d(L'), \rho(\tau))$, kjer je τ tip od x. Po drugi strani je $\rho(\tau) < d(R)$, torej $d(R') \leq d(R)$.
- Redekse, ki pridejo od zamenjave R z R'. Situacija je spet podobna tisti pri Lemi 7.2.1: ti redeksi imajo stopnjo enako $\rho(\tau)$, kjer je τ tip R in velja $\rho(\tau) < \rho(R)$.

□

Naslednji korak dokaza šibkega normalizacijskega izreka je definicija redukcije maksimalne stopnje, ki bo omogočala, da bo stopnja pri vsaki redukciji manjša od stopnje izraza pred redukcijo.

Lema 7.2.4. *Naj bo R redeks maksimalne stopnje n iz M. Predpostavimo, da imajo vsi redeksi vsebovani v R stopnjo manj kot n. Če dobimo M' iz M z zamenjavo R s R', potem ima M' striktno manjšo stopnjo od n.*

Dokaz. Ob zamenjavi se zgodi sledeče:

- Redeksi izven R ostanejo takšni kot so.
- Redeksi, ki so znotraj R so v splošnem ohranjeni, vendar lahko tudi pomnoženi: na primer če zamenjamo $(\lambda x.x, x)\, s$ z $\langle s, s\rangle$ se redeksi s pomnožijo. Tudi v tem primeru ostane stopnja manjša od n.

- Redeks R je reduciran in zamenjan z redeksi, ki imajo striktno manjšo stopnjo.

$\square$

Zdaj lahko sestavimo dokaz šibkega normalizacijskega izreka na osnovi prej prestavljenih lem.

Dokaz. (*Šibki normalizacijski izrek*) Naj bo M izraz in funkcija $\mu(M) = (n, m)$, kjer je

$$\begin{array}{rcl} n & = & d(M), \\ m & = & \text{število redeksov stopnje } n. \end{array}$$

Lema 7.2.4 pravi, da lahko iz M izberemo redeks R tako, da po konverziji $R \rightarrow_\beta R'$ s katero dobimo $M \rightarrow_\beta M'$ velja $\mu(M') \leq \mu(M)$: če $\mu(M') = (n', m')$ potem $n' < n$ ali $n' = n \wedge m' \leq m$.

Šibko normalizacijo M dokažemo z indukcijo po izpeljavi M. Na vsakem koraku predpostavimo, da lastnost velja za pod-izraze. Indukcijski korak pokažemo z uporabo Leme 7.2.4. $\square$

7.2.2 Strog normalizacijski izrek

Konstrukcija strogega normalizacijskega izreka je sledeča.

1. Najprej definiramo množico vseh λ-izrazov SN, ki strogo normalizirajo.

2. Nato definiramo *zasičene* množice $X \subseteq SN$, ki vsebujejo vse λ-izraze nad dano množico spremenljivk, ki strogo normalizirajo.

3. Pokažemo, da sta zasičeni množici SN in $[\![\alpha]\!]$ (vsi primerki tipa α) zasičeni in predstavimo operacije za katere je zasičenje zaprto.

4. Definiramo operacijo *ovrednotenja spremenljivk*, ki naredijo substitucijo prostih spremenljivk v λ-izrazih oz. v kontekstu Γ. Ovrednotenje ustreza delni evaluaciji λ-izraza.

5. Pokažemo, da ovrednotenje spremenljivk v λ-izrazih vedno vodi do elementov zasičenih množic, so SN.

Definicija 7.2.2. *1.* $SN = \{M \in \Lambda | M \text{ *strogo normalizira*} \}$

2. Naj $A, B \subseteq \Lambda$*.* $A \rightarrow B \subseteq \Lambda$ *definiramo*

$$A \rightarrow B = \{F \in \Lambda | \forall a \in A \; F\, a \in B\}.$$

3. *Za vsak* $\sigma \in Type(\lambda_{\to})$ *definiramo* $[\![\sigma]\!] \subseteq \Lambda$ *z naslednjimi pravili.*

$$\begin{array}{rcl} [\![\alpha]\!] & = & SN, \text{ kjer je } \alpha \text{ spremenljivka tipa} \\ [\![\sigma \to \tau]\!] & = & [\![\sigma]\!] \to [\![\tau]\!]. \end{array}$$

Definicija 7.2.3. 1. *Podmnožico* $X \subseteq SN$ *imenujemo* zasičeno, *če*

a) $\forall n \geq 0\, \forall R_1, \ldots, R_n \in SN : x\, \vec{R} \in X$, *kjer je* x *poljubna spremenljivka izraza.*

b) $\forall n \geq 0\, \forall R_1, \ldots, R_n \in SN\, \forall Q \in SN :$

$$[Q/x]P\, \vec{R} \in X \Rightarrow (\lambda x.P)\, Q\, \vec{R} \in X$$

2. $SAT = \{X \subseteq \Lambda | X \text{ je zasičena}\}$

Opazka 7.2.2. *Zasičene množice so SN po definiciji. Definicija zasičenih množic predstavlja postopek za enumeracijo izrazov zasičene množice. Konstruiramo jih od enostavnih izrazov proti bolj kompleksnim izrazom.*

Pravilo (1a) doseže, da so elementi zasičene množice X *vse spremenljivke* x *v primeru* $n = 0$.

Pogoj (1a) dodaja spremenljivke na začetek izraza, ki mu sledijo vsi možni izrazi SN. Pogoj (1b) dodaja k množici za vsak kontraktum še redeks, ki vodi do kontraktuma.

Z (1b) torej širimo množico izrazov, ki po vsaki uporabi pravila dobi množico novih izrazov z dodatnim redeksom ter vsemi možnimi izrazi SN na desni strani. □

Lema 7.2.5. 1. $SN \in SAT$

2. $A, B \in SAT \Rightarrow A \to B \in SAT$

3. *Naj bo* $\{A_i\}_{i \in I}$ *množica elementov* SAT, *potem* $\bigcap_{i \in I} A_i \in SAT$.

4. *Za vse* $\sigma \in type(\lambda_{\to})$ *velja* $[\![\sigma]\!] \in SAT$.

Dokaz. 1. SN zadošča pogoju (1a) definicije zasičenosti. Poglejmo še pogoj (1b). Predpostavljamo

$$[Q/x]P\, \vec{R} \in SN \wedge Q, \vec{R} \in SN. \tag{1}$$

Trdimo, da velja tudi

$$(\lambda x.P)\, Q\, \vec{R} \in SN. \tag{2}$$

Redukcije v P, Q in $\vec{R}$ se ustavijo, ker so po predpostavki SN. Izraz $[Q/x]P$ je pod-izraz izraza v SN torej je tudi sam SN, kot tudi P. Po končnem številu korakov redukcije izraza (2) dobimo $(\lambda x.P')\, Q'\, \vec{R}'$, kjer velja $P \rightarrow_{\beta}^{*} P'$, itd. Kontrakcija $(\lambda x.P')\, Q'\, \vec{R}'$ da

$$[Q'/x]P'\, \vec{R}'. \tag{3}$$

Ta izraz pa lahko vidimo kot reduciran izraz $[Q/x]P\, \vec{R}$. Ker je ta izraz SN potem je SN tudi (3) in $(\lambda x.P)\, Q\, \vec{R}$.

2. Naj bosta $A, B \in SAT$. Po definiciji velja $x \in A$ za vse spremenljivke. Potem

$$\begin{array}{rcll} F \in A \rightarrow B & \Rightarrow & F\, x \in B & \text{Po def.F} \\ & \Rightarrow & F\, x \in SN & \text{Po def.}SAT \\ & \Rightarrow & F \in SN & \text{Komponenta no} \end{array}$$

Velja torej $A \rightarrow B \subseteq SN$. $A \rightarrow B$ predstavlja vse funkcije iz A v B, ki so SN.

Poglejmo še pogoj (a) za zasičenost. Naj bo $\vec{R} \in SN$. Pokazati moramo, da $x\, \vec{R} \in A \rightarrow B$. To pomeni

$$\forall Q \in A : x\, \vec{R}\, Q \in B,$$

kar je res ker $A \subseteq SN$ in B je zasičena. Podobno sklepamo za pogoj zasičenosti (1b).

3. Podobno kot (2).

4. Z indukcijo na tvorbo σ, z uporabo (1) in (2).

□

Opazka 7.2.3. *Komentarji k Lemi 7.2.5.*

1. *$SN \in SAT$ pomeni, da je SN zasičena in vsebuje vse možne izraze, ki se jih da konstruirati s pravili (a) in (b) za zasičene množice.*

2. *Če sta zasičeni množici A in B potem je zasičena tudi $A \rightarrow B$. To pomeni, da je funkcija nad zasičenimi množicami vedno zasičena in tudi SN.*

3. *Nasičenje je zaprto za presek.*

4. *Za dano spremenljivko tipa α vsebuje $[\![\alpha]\!]$ vse SN λ-izraze, ki vsebujejo α. Ker je $[\![\alpha]\!]$ zasičena množica vsebuje vse izraze, ki se jih da generirati s pravili (1a) in (1b) Definicije 7.2.3.* □

Definicija 7.2.4. *1.* Ovrednotenje *v Λ je preslikava $\rho : V \rightarrow \Lambda$ kjer je V množica spremenljivk.*

2. *Naj bo ρ ovrednotenje v Λ. Potem*

$$[\![M]\!]_\rho = [\rho(x_1)/x_1, \ldots, \rho(x_n)/x_n]M,$$

kjer je $\vec{x} = x_1, \ldots, x_n$ množica prostih spremenljivk M.

Naj bo ρ ovrednotenje v Λ. Potem ρ reši *$M : \sigma$ zapisano $\rho \models M : \sigma$, če $[\![M]\!]_\rho \in [\![\sigma]\!]$.*

3. *Če je Γ kontekst, potem ρ reši Γ kar napišemo $\rho \models \Gamma$, če $\rho \models x : \sigma$ za vse $(x : \sigma) \in \Gamma$.*

4. *Kontekst Γ reši $M : \sigma$ kar zapišemo $\Gamma \models M : \sigma$, če*

$$\forall \rho[\rho \models \Gamma \Rightarrow \rho \models M : \sigma].$$

Opazka 7.2.4. 1. *Ovrednotenje priredi vsaki spremenljivki λ-izraz.*

2. *Ovrednotenje ρ izraza M predstavlja substitucijo vseh prostih spremenljivk x iz M z $\rho(x)$. Ovrednotenje potem predstavlja način evaluacije λ-izraza.*

Ovrednotenje ρ je rešitev $M : \sigma$ ($\rho \models M : \sigma$), če je rešitev M z ρ $[\![M]\!]_\rho$ element $[\![\sigma]\!]$ in je torej SN!

3. *Rešitev konteksta Γ z ovrednotenjem ρ ($\rho \models \Gamma$) predstavlja ovrednotenje spremenljivk $x \in \Gamma$ z $\rho(x)$. Spremenljivkam dodelimo izraze ($\in \Lambda$), začetne vrednosti na osnovi katerih lahko ovrednotimo izraze.*

4. *Kontekst Γ je rešitev $M : \sigma$ ($\Gamma \models M : \sigma$), če vsako ovrednotenje, ki reši Γ reši tudi $M : \sigma$.*

Γ torej reši $M : \sigma$, če za vsako prireditev SN izrazov spremenljivkam Γ pomeni, da je tudi $M : \sigma$ SN. □

Lema 7.2.6 (Uglašenost)**.**

$$\Gamma \vdash_{\lambda_\to} M : \sigma \Rightarrow \Gamma \models M : \sigma.$$

Dokaz. Z indukcijo po izpeljavi $M : \sigma$.

Primer 1. $\Gamma \vdash M : \sigma$ kjer je $M \equiv x$ sledi iz $(x : \sigma) \in \Gamma$. Potem je trivialno $\Gamma \models M : \sigma$.

Primer 2. $\Gamma \vdash M : \sigma$ kjer je $M \equiv M_1\, M_2$ je posledica $\Gamma \vdash M_1 : \tau \to \sigma$ in $\Gamma \vdash M_2 : \tau$.

Predpostavimo $\rho \models \Gamma$, da bi pokazali $\rho \models M_1\, M_2 : \sigma$. Potem po induktvni predpostavki velja $\rho \models M_1 : \tau \to \sigma$ in $\rho \models M_2 : \tau$ ali $[\![M_1]\!]_\rho \in [\![\tau \to \sigma]\!] = [\![\tau]\!] \to [\![\sigma]\!]$ in $[\![M_2]\!]_\rho \in [\![\tau]\!]$. Iz tega sledi $[\![M_1\, M_2]\!]_\rho = [\![M_1]\!]_\rho [\![M_2]\!]_\rho \in [\![\sigma]\!]$ ali $\rho \models M_1\, M_2 : \sigma$.

Primer 3. $\Gamma \vdash M : \sigma$ kjer je $M \equiv \lambda x.M'$ in $\sigma \equiv \sigma_1 \to \sigma_2$ je direktna posledica $\Gamma, x : \sigma_1 \vdash M' : \sigma_2$.

Po induktivni hipotezi velja

$$\Gamma, x : \sigma_1 \models M' : \sigma_2. \tag{1}$$

Predpostavimo $\rho \models \Gamma$, da bi pokazali $\rho \models (\lambda x.M') : \sigma_1 \to \sigma_2$. Drugače zapisano moramo pokazati

$$\forall N \in [\![\sigma_1]\!] : [\![\lambda x.M']\!]_\rho \, N \in [\![\sigma_2]\!].$$

Predpostavimo torej $N \in [\![\sigma_1]\!]$. Potem $\rho(N/x) \models \Gamma, x : \sigma_1$ in zato

$$[\![M']\!]_{\rho(N/x)} \in [\![\sigma_2]\!]$$

zaradi (1). Ker velja

$$\begin{array}{rcl} [\![\lambda x.M']\!]_\rho N & \equiv & [\rho(\vec{y})/\vec{y}](\lambda x.M')\, N \\ & \to_\beta^* & [\rho(\vec{y})/\vec{y}, N/x]M' \\ & \equiv & [\![M']\!]_{\rho(N/x)}, \end{array}$$

sledi zaradi zasičenosti $[\![\sigma_2]\!]$ da $[\![\lambda x.M']\!]_\rho N \in [\![\sigma_2]\!]$.

□

Izrek 7.2.3 (Strog normalizacijski izrek)**.** *Naj bo* $\Gamma \vdash_{\lambda_\rightarrow} M : \sigma$*. Potem* M *strogo normalizira.*

Dokaz. Naj bo $\Gamma \vdash M : \sigma$, potem velja $\Gamma \models M : \sigma$. Definiramo $\rho_0(x) = x$ za vsak x. Potem velja $\rho_0 \models \Gamma$, ker zaradi nasičenosti $[\![\tau]\!]$ za vsak x velja $x \in [\![\tau]\!]$. Zaradi tega velja $\rho_0 \models M : \sigma$ in zato $M \equiv [\![M]\!]_{\rho_0} \in [\![\sigma]\!] \subseteq SN$. □

7.3 Curry-Howardov izomorfizem

Curry-Howardov izomorfizem prikaže sintaktično in pomensko zvezo med logiko in teorijo tipov. V tej sekciji si bomo ogledali najbolj osnovno zvezo, ki definirana med *izjavnim računom* in $lambda$-*računom s tipi* ($\lambda_\rightarrow$). Kasneje bodo omenjene še druge zveze, ki so definirane nad jeziki z večjo izrazno močjo.

Definirali bomo preslikavo med izrazi logike in izrazi $\lambda_\rightarrow$ tako, da bo deduktivno sklepanje na strani logike preslikano v normalizacijo izrazov λ-računa. Hkrati bomo definirali tudi preslikavo v nasprotno smer.

Dokazi izjavnega računa ustrezajo ovrednotenju izrazov v λ-računu. Preslikava torej ni samo bijektivna ampak tudi ohranja preslikave pri deduktivnem sklepanju na eni strani oz. pri normalizaciji izrazov na drugi, analogni strani.

7.3.1 Definicija izomorfizma

Na eni strani imamo dokaze izjav napisanih v izjavnem računu. Za dokazovanje uporabljamo naravno dedukcijo, ki je bila predstavljena v Poglavju 2.

Dokaze v izjavnem računu bomo videli kot izpeljave izrazov $\lambda_\rightarrow$. Bolj natančno, dokaz izjave A postane v $\lambda_\rightarrow$ izpeljava izraza tipa A.

Uporaba tipov $\times, \rightarrow$ bo ustrezala logičnim operacijam $\wedge, \Rightarrow$.

Na drugi strani imamo izraze $\lambda_\rightarrow$, ki imajo tipe. Evaluacija izraza bo ustrezala sklepanju v izjavnem računu.

Poglejmo si zdaj pravila, ki definirajo bijekcijo med izrazi $\lambda_\rightarrow$ in izpeljavami dokazov izjavne logike.

1. Dedukcija A ustreza spremenljivki x_i^A.
2. Dedukcija $\dfrac{\begin{matrix}\vdots \\ A\end{matrix} \quad \begin{matrix}\vdots \\ B\end{matrix}}{A \wedge B}\wedge\mathcal{I}$ ustreza izrazu $\langle u, v\rangle$, kjer u ustreza izpeljavi A in v izpeljavi B.
3. Dedukciji $\dfrac{\begin{matrix}\vdots \\ A \wedge B\end{matrix}}{A}\wedge 1\mathcal{E}$ in $\dfrac{\begin{matrix}\vdots \\ A \wedge B\end{matrix}}{B}\wedge 2\mathcal{E}$ ustrezata izrazoma $\Pi^1 t$ in $\Pi^2 t$, kjer t ustreza dedukciji $A \wedge B$.
4. Dedukcija $\dfrac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{A \Rightarrow B}\Rightarrow\mathcal{I}$ ustreza izrazu $\lambda x_i^A.v$, kjer izpeljava A ustreza predpostavkam za izpeljavo x_i^A ter izpeljava B iz A ustreza ovrednotenju v.

5. Dedukcija $\dfrac{\begin{matrix}\vdots \\ A\end{matrix} \qquad \begin{matrix}\vdots \\ A \Rightarrow B\end{matrix}}{B} \Rightarrow \mathcal{E}$ ustreza izrazu $t\ u$, kjer t in u ustrezata izpeljavama $A \Rightarrow B$ in B.

7.3.2 Pomen izomorfizma

7.4 Opombe

Lambda račun s tipi je podrobno predstavljen v članku Barendregt [4]. Nekatere lastnosti lambda računa s tipi so podane v knjigi Barendregt [2].

Noramalizacija izrazov lambda računa s tipi je predstavljena v knjigi Barendregt [2]. Povzetek dokazov šibkega in močnega normalizacijskega izreka za λ-račun s tipi je predstavljen v knjigi Girarda [5].

Različne verzije dokaza normalizacijskega izreka za lambda račun s tipi se nahajajo v člankih avtorjev Hindley [], Berger [], Filinski [], ...