

Izpeljava tipov

Contents

- [Primer](#)
- [Nastavljanje enačb](#)
- [Reševanje enačb](#)
- [Lastnosti Hindley-Milnerjevega algoritma](#)

Videli smo, da za izraze, ki jim lahko priredimo tip, velja izrek o varnosti, kmalu pa bomo videli, da jih lahko interpretiramo z obstoječimi matematičnimi koncepti. Vendar je ročno prirejanje tipov zamudno, zato bi radi, da to namesto nas naredi računalnik. V ta namen bomo spoznali Hindley-Milnerjev algoritem, ki izpelje najbolj splošen tip, ki ga lahko priredimo izrazu.

Hindley-Milnerjev algoritem poteka v dveh fazah. Najprej se rekurzivno sprehodi čez izraz in določi tipe tam, kjer je to očitno (na primer `42` bo tipa `int`), na preostalih mestih pa pusti neznanke, ki morajo zadoščati določenim enačbam. Na primer, vemo, da bo aplikacija $M\ N$ nekega tipa B , če bo M tipa $A \rightarrow B$ in bo N tipa A . V ta namen sintakso tipov razširimo z neznankami α , ki jih zaradi razločevanja od že znanih spremenljivk x imenujemo *parametri*. V drugi fazi dobljene enačbe reši.

$$A, B ::= \alpha \mid \mathbf{bool} \mid \mathbf{int} \mid A \rightarrow B$$

Primer

Da dobimo intuicijo, poskusimo izpeljati tip izraza $M = \lambda f. \lambda x. f(f\ x)$. Ker je M funkcija, bo njen tip funkcijski. Problem lahko prevedemo na manjšega, če gremo izračunati tip telesa $\lambda x. f(f\ x)$, vendar moramo pred tem vedeti, kakšen tip bomo priredili spremenljivki f . Ker tega na tej točki še ne vemo, si izberemo nek svež parameter α . Ko bomo izračunali, da je tip telesa recimo B , bomo vedeli tudi, da je tip celotnega izraza M enak $\alpha \rightarrow B$. Ker je tudi telo funkcija, podoben korak ponovimo še enkrat ter se ob dodatni predpostavki $x : \beta$ lotimo računanja tipa $f(f\ x)$. Vse skupaj lahko predstavimo z drevesom

$$\frac{\frac{\frac{\Gamma \vdash f : \alpha}{\Gamma \vdash f : \alpha} \quad \frac{\Gamma \vdash f : \alpha \quad \Gamma \vdash x : \beta}{\Gamma \vdash f\ x : \gamma} \quad (\alpha = \beta \rightarrow \gamma)}{\underbrace{f : \alpha, x : \beta \vdash f(f\ x) : \delta}_{\Upsilon}} \quad (\alpha = \gamma \rightarrow \delta)}{\frac{f : \alpha \vdash \lambda x. f(f\ x) : \beta \rightarrow \delta}{\vdash \lambda f. \lambda x. f(f\ x) : \alpha \rightarrow (\beta \rightarrow \delta)}}$$

Na primer, ko želimo izračunati tip aplikacije $f\ x$, vemo, da zaradi predpostavk v kontekstu velja $f : \alpha$ in $x : \beta$. Po drugi strani pa mora biti f funkcija, ki sprejema argumente tipa β , torej velja $\alpha = \beta \rightarrow \gamma$ za nek neznani tip γ , ki je tudi tip aplikacije. Podobno lahko ob predpostavki $\alpha = \gamma \rightarrow \delta$ izračunamo, da je tip telesa funkcije $f(f\ x)$ enak δ .

Če si ogledamo dobljeni enačbi $\alpha = \beta \rightarrow \gamma = \gamma \rightarrow \delta$ hitro vidimo še, da velja $\beta = \gamma = \delta$, zato je tip izraza M enak $(\beta \rightarrow \beta) \rightarrow (\beta \rightarrow \beta)$, in to je tudi najbolj splošen tip.

Nastavljanje enačb

Določanje sistema enačb opišemo z relacijo oblike $\Gamma \vdash M : A \mid \mathcal{E}$, ki pravi, da ima v kontekstu Γ izraz M tip A , če veljajo enačbe \mathcal{E} , predstavljene s seznamom oblike $A_1 = B_1, \dots, A_n = B_n$. Relacijo podamo s pravili, ki so podobna pravilom za izpeljavo tipov, le da je tip podizrazov v predpostavkah poljuben, enakosti med njimi pa zapišemo v \mathcal{E} . Na primer, pravilo za določanje tipa aplikacije je:

$$\frac{\Gamma \vdash M_1 : A \rightarrow B \quad \Gamma \vdash M_2 : A}{\Gamma \vdash M_1\ M_2 : B}$$

kjer vidimo, da je moral biti tip M_1 funkcijski z domeno enako tipu od M_2 . Pravilo za izpeljavo tipa pa je

kjer za podizraza predpostavimo poljubna tipa A_1 in A_2 ne zahtevamo ničesar, ujemanje med njima pa zapišemo v enačbo $A_1 = A_2 \rightarrow \alpha$ v zaključku pravila. Poleg te enačbe bo morala vsaka rešitev zadostiti vsem enačbam \mathcal{E}_1 in \mathcal{E}_2 , ki jih ravno tako dodamo v zaključek.

Missing \end{split}

Pri tem upoštevamo, da morajo biti vsi parametri α sveži, torej da jih poprej še nismo uporabili. Drevo izpeljave za izraz $\lambda f. \lambda x. f(f x)$ od pravega je natančneje torej:

Ker pa je v pravilih množico enačb \mathcal{E} le povečujemo, jo v praksi samo napišemo na stran in tako izpeljavo bolj ekonomično napišemo kot:

Reševanje enačb

Rešitev sistema enačb bomo predstavili s *substitucijo* $\sigma = \alpha_1 \mapsto A_1, \dots, \alpha_n \mapsto A_n$, ki pove, da je vrednost parametra α_i enaka A_i . Dano substitucijo σ lahko uporabimo na poljubnem tipu A , da dobimo tip $\sigma(A)$, ki je rekurzivno definiran kot:

Definiramo lahko tudi kompozitum $\sigma \circ \sigma'$, ki vse parametre α , za katere obstaja $(\alpha \mapsto A)$ in σ' , slika v $\sigma(A)$, vse preostale parametre pa tako σ . Hitro lahko preverimo, da velja $(\alpha \mapsto A)(B) = B[A/\alpha]$ ter $(\sigma \circ \sigma')(A) = \sigma(\sigma'(A))$.

Pravimo, da substitucija σ reši množico enačb \mathcal{E} , kar pišemo kot $\sigma \models \mathcal{E}$, kadar sta za vsako enačbo $(A = B) \in \mathcal{E}$ tipa $\sigma(A)$ in $\sigma(B)$ enaka.

Najbolj splošno rešitev množice enačb \mathcal{E} dobimo tako, da postopoma razrešujemo enačbe v njej. Če sta tipa v prvi enačbi že enaka, se je lahko znebite. Če sta oba funkcijska, enačbo razbijemo na dve enačbi med domenama in kodomenama. Zadnji primer, kjer rešitev obstaja, pa je ta, da na eni strani nastopa parameter in je oblike $\alpha = A$ (ali obratno). Načeloma to pomeni, da bomo α substituirali z A , težavo imamo le, kadar A vsebuje α . V tem primeru enačba ne more imeti rešitve. Formalno reševanje podamo z relacijo $\mathcal{E} \searrow \sigma$, podano s pravili:

$$\frac{(\alpha \mapsto A)(\mathcal{E}) \searrow \sigma \quad \alpha \notin fv(A)}{(\alpha \mapsto A)(\mathcal{E} \searrow \sigma) \searrow \sigma} \quad \text{(\textit{A} \searrow)}$$

$$A = \alpha, \mathcal{E} \searrow (\alpha \mapsto \sigma(A)) \circ \sigma$$

kjer je množica prostih parametrov $fv(A)$ definirana kot $fv(\alpha) = \{\alpha\}$ $fv(\mathbf{bool}) = fv(\mathbf{int}) = \emptyset$ $fv(A \rightarrow B) = fv(A) \cup fv(B)$

Lastnosti Hindley-Milnerjevega algoritma

Da Hindley-Milnerjev algoritem res vrne najbolj splošen tip, pokažemo s sledečimi trditvami.

Pravilnost nastavljanja enačb

Najprej pokažemo, da vsaka rešitev dobljenih enačb res predstavlja veljavno prirejanje tipa.

Trditev. Če velja $\Gamma \vdash M : A \mid \mathcal{E}$ in $\sigma \models \mathcal{E}$, potem $\sigma(\Gamma) \vdash M : \sigma(A)$.

Dokaz. Z indukcijo na $\Gamma \vdash M : A \mid \mathcal{E}$. Za primer si oglejmo pravilo za aplikacijo

$$\frac{\Gamma \vdash M_1 : A_1 \mid \mathcal{E}_1 \quad \Gamma \vdash M_2 : A_2 \mid \mathcal{E}_2}{\Gamma \vdash M_1 M_2 : \alpha \mid A_1 = A_2 \rightarrow \alpha, \mathcal{E}_1, \mathcal{E}_2}$$

Če velja $\sigma \models A_1 = A_2 \rightarrow \alpha, \mathcal{E}_1, \mathcal{E}_2$, potem velja tudi $\sigma \models \mathcal{E}_1$ in $\sigma \models \mathcal{E}_2$, saj gre za manjšo množico enačb. Po indukcijski predpostavki potem velja $\sigma(\Gamma) \vdash M_1 : \sigma(A_1)$ in $\sigma(\Gamma) \vdash M_2 : \sigma(A_2)$. Hkrati velja $\sigma(A_1) = \sigma(A_2) \rightarrow \sigma(\alpha)$, zato velja $\sigma(\Gamma) \vdash M_1 : \sigma(\alpha) \rightarrow \sigma(A_2)$, zato lahko uporabimo pravilo za določanje tipa aplikacije, da dobimo

$$\frac{\sigma(\Gamma) \vdash M_1 : \sigma(A_2) \rightarrow \sigma(\alpha) \quad \sigma(\Gamma) \vdash M_2 : \sigma(A_2)}{\sigma(\Gamma) \vdash M_1 M_2 : \sigma(\alpha)}$$

kar je natanko to, kar smo želeli. ■

Pravilnost reševanja enačb

Prav tako moramo pokazati pravilnost druge faze.

Trditev. Če velja $\mathcal{E} \searrow \sigma$, potem velja $\sigma \models \mathcal{E}$.

Dokaz. Tudi tu uporabimo indukcijo na $\mathcal{E} \searrow \sigma$. Za primer vzemimo pravilo

$$\frac{(\alpha \mapsto A)(\mathcal{E}) \searrow \sigma \quad \alpha \notin fv(A)}{\alpha = A, \mathcal{E} \searrow (\alpha \mapsto \sigma(A)) \circ \sigma}$$

Najprej se spomnimo, da velja $(\alpha \mapsto A)(\mathcal{E}) = \mathcal{E}[A/\alpha]$. Ker v množici enačb $\mathcal{E}[A/\alpha]$ parameter α ne nastopa, lahko pokažemo, da tudi v σ ni omenjen, zato velja $\sigma(\alpha) = \alpha$.

Tako v prvi enačbi na levi strani dobimo $((\alpha \mapsto \sigma(A)) \circ \sigma)(\alpha) = (\alpha \mapsto \sigma(A))(\sigma(\alpha)) = (\alpha \mapsto \sigma(A))(\alpha) = \sigma(A)$ *nadesnistranipa*
 $((\alpha \mapsto \sigma(A)) \circ \sigma)(A) = (\alpha \mapsto \sigma(A))(\sigma(A)) = \sigma(A)$ *saj* $\alpha \notin fv(\sigma(A))$, *ker* $\alpha \notin fv(A)$, *vs substituciji* α pa se prav tako ne pojavi.

Dokazati moramo še $(\alpha \mapsto \sigma(A)) \circ \sigma \models \mathcal{E}$. Vzemimo torej poljubno enačbo $A_1 = A_2 \in \mathcal{E}$ in pokažimo, da velja $(\alpha \mapsto \sigma(A))(\sigma(A_1)) = (\alpha \mapsto \sigma(A))(\sigma(A_2))$ oziroma $\sigma(A_1)[\sigma(A)/\alpha] = \sigma(A_2)[\sigma(A)/\alpha]$. Po indukcijski predpostavki velja $\sigma \models (\alpha \mapsto A)(\mathcal{E})$, iz česar sledi $\sigma(A_1[A/\alpha]) = \sigma(A_2[A/\alpha])$, kar je natanko to, kar moramo pokazati, saj velja $\sigma(A_1)[\sigma(A)/\alpha] = \sigma(A_1[A/\alpha])$ in podobno za A_2 . ■

Polnost nastavljanja enačb

Pokazati moramo še, da so nastavljene enačbe najbolj splošne, torej da lahko vsak tip, ki bi ga lahko priredili danemu izrazu, dobimo kot konkretno substitucijo izpeljanega tipa. Najprej pokažemo, da za vsak izraz s tipom lahko nastavimo sistem enačb:

Trditev. Naj bo $\Gamma \vdash M : A$ Tedaj obstaja A' in \mathcal{E} , da velja $\Gamma \vdash M : A' \mid \mathcal{E}$.

Dokaz. Dokaz je rutinska indukcija, saj so pravila za $\Gamma \vdash M : A' \mid \mathcal{E}$ tako splošna, da jih lahko uporabimo na poljubnem izrazu. Paziti moramo le na to, da v katerem se vse proste spremenljivke pojavijo v Γ , kar pa nam zagotavlja predpostavka $\Gamma \vdash M : A$. ■

Ko vemo, da enačbe lahko vedno nastavimo, na lahko novemo tudi, da so najbolj splošne:

ko vemo, da enačba lahko vedno nastavimo, pa lahko povelje tudi, da se najbolj splošno.

Trditev. Naj bo $\Gamma \vdash M : A \mid \mathcal{E}$. Tedaj za poljuben tip A' , za katerega velja in $\Gamma \vdash M : A$, obstaja $\sigma \models \mathcal{E}$, da velja $\sigma(A) = A'$.

Dokaza ne bomo navajali, ker je bolj tehničen, najdete pa ga lahko v razdelku 22.5 učbenika [Types and programming languages](#).

Polnost reševanja enačb

Tudi v drugi fazi moramo pokazati, da je najdena rešitev res najbolj splošna. Tudi tu najprej pokažemo, da rešitev vedno obstaja.

Trditev. Če velja $\sigma \models \mathcal{E}$, tedaj obstaja σ' , da velja $\mathcal{E} \searrow \sigma'$.

Dokaz. Dokaz poteka z malo bolj zapleteno indukcijo, saj ni očitno, katera stvar se v predpostavkah manjša. Na primer, včasih se enačb znebimo, včasih zamenjamo eno enačbo z več novimi, vendar z manjšimi tipi, včasih pa tipe povečamo, vendar se znebimo parametrov. Vse to zajamemo z leksikografsko urejenostjo med sistemi enačb.

Definirajmo velikost $|A|$ tipa A kot $|\alpha| = |\mathbf{bool}| = |\mathbf{int}| = 1 \quad |A \rightarrow B| = 1 + |A| + |B|$
 $|\mathcal{E}| = \sum_{(A_1=A_2) \in \mathcal{E}} (|A_1| + |A_2|)$
 $fv(\mathcal{E}) = \bigcup_{(A_1=A_2) \in \mathcal{E}} fv(A_1) \cup fv(A_2)$

Vzemimo zdaj enačbe \mathcal{E} in nadaljujmo z leksikografsko indukcijo na $(|fv(\mathcal{E})|, |\mathcal{E}|)$. Torej, na vsakem koraku se bo število prostih parametrov zmanjšalo, ali pa bo ostalo enako, zmanjšala pa se bo skupna velikost vseh tipov v enačbah.

Če je $\mathcal{E} = \emptyset$, velja $\mathcal{E} \searrow \emptyset$. V nasprotnem primeru imamo $\mathcal{E} = (A_1 = A_2, \mathcal{E}')$, kjer velja $\sigma \models \mathcal{E}'$ in $\sigma(A_1) = \sigma(A_2)$. Poglejmo, v kakšnih primerih se slednje lahko zgodi.

Če je $A_1 = A_2$, lahko uporabimo pravilo $\frac{\mathcal{E}' \searrow \sigma'}{A_1 = A_1, \mathcal{E}' \searrow \sigma'} \text{ sa } j \text{ po indukciji obstaja } \sigma' \models \mathcal{E}'$.

Če pa je $A_1 \neq A_2$, hkrati pa $\sigma(A_1) = \sigma(A_2)$, pa morata biti A_1 in A_2 vsaj kompatibilne oblike. Torej sta oba funkcijska tipa, ali pa je eden od njiju parameter.

V prvem primeru je $A_1 = A'_1 \rightarrow A''_1$ in $A_2 = A'_2 \rightarrow A''_2$, zato definiramo $\mathcal{E}'' = (A'_1 = A'_2, A''_1 = A''_2, \mathcal{E}')$. Ker velja $fv(\mathcal{E}'') = fv(\mathcal{E})$ in $|\mathcal{E}''| < |\mathcal{E}|$, hkrati pa velja $\sigma \models \mathcal{E}''$, po indukciji obstaja rešitev $\mathcal{E}'' \searrow \sigma'$, zato lahko uporabimo pravilo

$$\frac{\underbrace{A_1 = B_1, A_2 = B_2, \mathcal{E}' \searrow \sigma'}_{\mathcal{E}''}}{\underbrace{A_1 \rightarrow A_2 = B_1 \rightarrow B_2, \mathcal{E}' \searrow \sigma'}_{\mathcal{E}}}$$

V drugem primeru pa najprej pogledimo možnost, da je parameter na levi strani, torej $A_1 = \alpha$. Naj bo $\mathcal{E}'' = \mathcal{E}[A_2/\alpha]$. Tedaj je $|fv(\mathcal{E}'')| < |fv(\mathcal{E})|$, saj smo se α znebili, novih parametrov pa nismo dodajali, saj je $fv(A_2) \subseteq fv(\mathcal{E})$. Če velja $\sigma \models \mathcal{E}$, potem velja tudi $\sigma \models \mathcal{E}[A_2/\alpha]$, saj smo s substitucijo kvečjemu izenačili več tipov. Torej po indukciji obstaja rešitev $\mathcal{E}[A_2/\alpha] \searrow \sigma'$. Hkrati mora veljati $\alpha \notin fv(A_2)$, saj sicer ne more veljati $\sigma(\alpha) = \sigma(A_2)$, ker bo desna stran striktno večja od leve. Tako so izpolnjene vse predpostavke, da uporabimo drugo pravilo:

$$\frac{(\alpha \mapsto A_2)(\mathcal{E}) \searrow \sigma' \quad \alpha \notin fv(A_2)}{\alpha = A_2, \mathcal{E} \searrow (\alpha \mapsto \sigma(A_2)) \circ \sigma}$$

Če je parameter na desni strani, ravnamo simetrično. ■

Ko vemo, da rešitev obstaja, pokažemo še, da je najbolj splošna.

Trditev. Če velja $\mathcal{E} \searrow \sigma$, tedaj za poljubno $\sigma' \models \mathcal{E}$ obstaja σ'' , da velja $\sigma' = \sigma'' \circ \sigma$.

Dokaz. Dokaz poteka z indukcijo na $\mathcal{E} \searrow \sigma$. Za primer vzemimo pravilo $\frac{A_1 = B_1, A_2 = B_2, \mathcal{E} \searrow \sigma}{A_1 \rightarrow A_2 = B_1 \rightarrow B_2, \mathcal{E} \searrow \sigma}$. Naj velja $\sigma' \models A_1 \rightarrow A_2 = B_1 \rightarrow B_2$, $\sigma' \models \mathcal{E}$. Teda velja tudi $\sigma' \models A_1 = B_1, A_2 = B_2$, $\sigma' \models \mathcal{E}$ zato po indukciji obstaja $\sigma'' \models \mathcal{E}$, davelja $\sigma' = \sigma'' \circ \sigma$, kar je natanko to, kar smo želeli pokazati.