LTAT.03.019 Funktsionaalprogrammeerimine

Lihtsalt tüübitud λ-arvutus

Tüübisüsteemid

- Tüübid spetsifitseerivad programmide omadusi.
  - λ-arvutuses klassifitseerivad normaalkujusid.
  - Kui term on tüüpi Nat, ja ta on normaalkujul, siis see term esitab naturaalarvu.
- Tüübisüsteem koosneb kolmest komponendist:
  - tüüpide hulk;
  - programmide hulk;
  - tüüpimisreeglid.

Lihtsad tüübid

- Olgu antud loenduv hulk tüübimuutujaid (baastüübid).
- Lihtsad tüübid = ei ole polümorfsed.
- Lihtsate tüüpide süntaks:

$$\begin{array}{lcll} \tau & ::= & \alpha & \text{tüübimuutuja} \\ & | & \tau_1 \to \tau_2 & \text{funktsioonitüüp} \end{array}$$

- Tüübikonstruktor $\to$ on paremassotsiatiivne.

$$\tau_1 \to \tau_2 \to \tau_3 \equiv \tau_1 \to (\tau_2 \to \tau_3)$$

Church vs. Curry

- Tüüpidega λ-avaldiste esitamiseks on kaks peamist stiili.
- Church'i stiil: funktsiooni parameetrite tüübid esitatakse ilmutatult λ-termis $\lambda x : \tau.\ e$.
  - Näiteks: $\lambda x : \mathrm{Nat}.\ x + x : \mathrm{Nat} \to \mathrm{Nat}$.
  - Igal termil on unikaalne tüüp.
- Curry stiil: termid on tavalised (s.o. ilma tüüpideta) ja kasutatakse tüübikorrektsuse predikaati (tüübituletus).
  - Näide: $\lambda x.\ x + x : \mathrm{Nat} \to \mathrm{Nat}$.
  - Termidel on palju tüüpe (näit. $\lambda x.\ x : \mathrm{Nat} \to \mathrm{Nat}$, kuid ka $\lambda x.\ x : \mathrm{Bool} \to \mathrm{Bool}$).
- Meie kasutame põhiliselt esimest stiili.

Termid ja redutseerimine

- Termide süntaks:

$$\begin{array}{rcll} e & ::= & x & \text{muutuja} \\ & \mid & e_1\ e_2 & \text{aplikatsioon} \\ & \mid & \lambda x : \tau .\ e & \text{abstraktsioon} \end{array}$$

- Samad süntaktilised konventsioonid kui puhtas λ-arvutuses.
- Substitutsioon, reduktsioon jmt on defineertud analoogselt puhta λ-avutusega ignoreerides tüübiannotatsioone.

Tüüpimisrelatsioon

- Et teha kindlaks mis termid on millist tüüpi, defineerime tüüpimisrelatsiooni.
- Baasrelatsioon on kujul $\Gamma \vdash e : \tau$
  - *"term e on tüüpi τ kontekstis Γ"*.
- Kontekst Γ on muutujast ja tüübist koosnevate paaride jada $\Gamma = \{x_1 : \tau_1, \ \ldots, \ x_n : \tau_n\}$.
  - Konteksti Γ doomenit tähistame $\mathrm{dom}(\Gamma) = \{x_1, \ \ldots, \ x_n\}$.
  - Korrektses tüüpimisrelatsioonis peab olema $\mathrm{FV}(e) \subseteq \mathrm{dom}(\Gamma)$.
- $\Gamma \leq \Delta$ tähendab, et kui $x \in \mathrm{dom}(\Gamma)$, siis $x \in \mathrm{dom}(\Delta)$ ja $\Gamma(x) = \Delta(x)$.
- Programm (s.o. kinnine term) e on tüüpi τ, kui tal on see tüüp tühjas kontekstis $\vdash e : \tau$.

Tüüpimisrelatsioon

- Tüüpimisrelatsioon on defineeritud tüüpimisreeglite abil.
- Lihtsalt tüübitud λ-arvutuse tüüpimisreeglid:

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}\ \mathit{var} \qquad \frac{\Gamma, x : \sigma \vdash e : \tau}{\Gamma \vdash \lambda x : \sigma .\, e : \sigma \to \tau}\ \mathit{abs}$$

$$\frac{\Gamma \vdash e_1 : \sigma \to \tau \quad \Gamma \vdash e_2 : \sigma}{\Gamma \vdash e_1\, e_2 : \tau}\ \mathit{app}$$

  kus $x \notin \mathrm{dom}(\Gamma)$.
- Edaspidi tähistame lihtsalt tüübitud λ-arvutust $\lambda\!\to$.

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (1):

$$\frac{}{\vdash \lambda x{:}\tau,\ y{:}\sigma.\, x :}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (1):

$$\dfrac{\overline{\{x{:}\tau\} \vdash \lambda y{:}\sigma.\, x :}}{\vdash \lambda x{:}\tau,\ y{:}\sigma.\, x : \tau \rightarrow}\ \textit{abs}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (1):

$$\cfrac{\cfrac{\overline{\{x{:}\tau,\ y{:}\sigma\} \vdash x :}}{\{x{:}\tau\} \vdash \lambda y{:}\sigma.\, x : \sigma \to}\ \mathit{abs}}{\vdash \lambda x{:}\tau,\ y{:}\sigma.\, x : \tau \to \sigma \to}\ \mathit{abs}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (1):

$$\dfrac{\dfrac{\overline{\{x{:}\tau,\ y{:}\sigma\} \vdash x : \tau}\ \textit{var}}{\{x{:}\tau\} \vdash \lambda y{:}\sigma.\, x : \sigma \to \tau}\ \textit{abs}}{\vdash \lambda x{:}\tau,\ y{:}\sigma.\, x : \tau \to \sigma \to \tau}\ \textit{abs}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (2):

$$
\dfrac{\dfrac{\dfrac{\Gamma \vdash x : \sigma \to \tau \qquad \dfrac{\Gamma \vdash y : \rho \to \sigma \quad \Gamma \vdash z : \rho}{\Gamma \vdash yz : \sigma}}{\Gamma \vdash x(yz) : \tau}}{\Gamma_2 \vdash \lambda z{:}\rho.\, x(yz) : \rho \to \tau}}{\dfrac{\Gamma_1 \vdash \lambda y{:}\rho\to\sigma,\ z{:}\rho.\, x(yz) : (\rho \to \sigma) \to \rho \to \tau}{\vdash \lambda x{:}\sigma\to\tau,\ y{:}\rho\to\sigma,\ z{:}\rho.\, x(yz) : (\sigma \to \tau) \to (\rho \to \sigma) \to \rho \to \tau}}
$$

kus

$$
\begin{array}{lcl}
\Gamma_1 & = & \{x{:}\sigma\to\tau\} \\
\Gamma_2 & = & \{x{:}\sigma\to\tau,\ y{:}\rho\to\sigma\} \\
\Gamma & = & \{x{:}\sigma\to\tau,\ y{:}\rho\to\sigma,\ z{:}\rho\}
\end{array}
$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\dfrac{\qquad\qquad\qquad\qquad}{\vdash \lambda x{:}\tau.\, x\, x :}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\dfrac{\dfrac{}{\{x{:}\tau\} \vdash x\,x :}}{\vdash \lambda x{:}\tau.\, x\,x : \tau \rightarrow}\ \mathit{abs}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\dfrac{\dfrac{\dfrac{}{\{x{:}\tau\} \vdash x : \sigma \to\ \ } \qquad \dfrac{}{\{x{:}\tau\} \vdash x : \sigma}}{\{x{:}\tau\} \vdash x\, x :\ \ }\ \textit{app}}{\vdash \lambda x{:}\tau.\, x\, x : \tau \to\ \ }\ \textit{abs}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\dfrac{\dfrac{\dfrac{}{\{x{:}\tau\} \vdash x : \tau \to} \qquad \dfrac{}{\{x{:}\tau\} \vdash x : \tau}\,\mathit{var}}{\{x{:}\tau\} \vdash x\,x :}\,\mathit{app}}{\vdash \lambda x{:}\tau.\, x\,x : \tau \to}\,\mathit{abs}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\dfrac{\dfrac{}{\{x{:}\tau\} \vdash x : \tau \to \rho}\ \mathit{var} \qquad \dfrac{}{\{x{:}\tau\} \vdash x : \tau}\ \mathit{var}}{\dfrac{\{x{:}\tau\} \vdash x\,x : \rho}{\vdash \lambda x{:}\tau.\, x\,x : \tau \to \rho}\ \mathit{abs}}\ \mathit{app}$$

Tüüpimispuu konstrueerimine ebaõnnestub!

Tüübiohutus

- Tüübiohutus:
  Korrektselt tüübitud programmid "ei lähe valesti".
- Hind:
  - Mõned mõistlikud programmid hüljatakse.
- Koosneb kahest osast:
  - Progress: Korrektselt tüübitud ei ole tupikus (ta on kas vöörtus või saab teha ühe väärtustusreeglitele vastava sammu).
  - Säilitamine: Kui korrektselt tüübitud term teeb ühe sammu, siis ka resultaat on korrektselt tüübitud.

Tüüpide unikaalsus, säilimine ja progress

- Tüüpide unikaalsus:
  Kui $\Gamma \vdash e : \tau_1$ ja $\Gamma \vdash e : \tau_2$, siis $\tau_1 = \tau_2$.
- NB! Tüüpide unikaalsus ei kehti mitmete rikkamate keelte korral.

- Tüüpide säilimine (subjekt-reduktsioon):
  Kui $\Gamma \vdash e : \tau$ ja $e \twoheadrightarrow_\beta e'$, siis $\Gamma \vdash e' : \tau$.

- Progress:
  Kui $\Gamma \vdash e : \sigma$, siis $\exists e' : \sigma.e \rightarrow_\beta e'$ või $e \in \mathrm{Val}$, kus

$$v \in \mathrm{Val} \ ::= \ x\, v \ldots v \mid \lambda x{:}\tau.v$$

Tugev normaliseerimine

- Tugev normaliseerimine:
  Lihtsalt tüübitud λ-arvutuses $\lambda\!\rightarrow$ on iga β-reduktsiooni jada lõplik.
- Mõned lihtsad järeldused:
  - Termide võrdsuse küsimus on lahenduv.
  - Püsipunktikombinaatorid ei ole defineeritavad.

Laiendus: tõeväärtused

- Tüübid: $\tau ::= \ldots \mid \mathrm{Bool}$
- Termid: $e ::= \ldots \mid \mathtt{true} \mid \mathtt{false} \mid \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2$
- Väärtused: $v ::= \ldots \mid \mathtt{true} \mid \mathtt{false}$
- Tüüpimisreeglid:

$$\frac{}{\Gamma \vdash \mathtt{true} : \mathrm{Bool}} \qquad \frac{}{\Gamma \vdash \mathtt{false} : \mathrm{Bool}}$$

$$\frac{\Gamma \vdash e_0 : \mathrm{Bool} \quad \Gamma \vdash e_1 : \tau \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash \mathsf{if}\ e_0\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \tau}$$

- Väärtustamisreeglid:

$$\begin{array}{lcl} \mathsf{if}\ \mathtt{true}\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 & \to & e_1 \\ \mathsf{if}\ \mathtt{false}\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 & \to & e_2 \end{array}$$

Laiendus: ühiktüüp

- Tüübid: $\tau ::= \ldots \mid \mathrm{Unit}$
- Termid: $e ::= \ldots \mid ()$
- Väärtused: $v ::= \ldots \mid ()$
- Tüüpimisreeglid:

$$\frac{}{\Gamma \vdash () : \mathrm{Unit}}$$

- Väärtustamisreegleid ei ole!

Laiendus: paarid

- Tüübid: $\tau ::= \ldots \mid \tau_1 \times \tau_2$
- Termid: $e ::= \ldots \mid (e_1, e_2) \mid \texttt{fst} \mid \texttt{snd}$
- Väärtused: $v ::= \ldots \mid (v_1, v_2)$
- Tüüpimisreeglid:

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (e_1, e_2) : \tau_1 \times \tau_2}$$

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \texttt{fst}\, e : \tau_1} \quad \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \texttt{snd}\, e : \tau_2}$$

- Väärtustamisreeglid:

$$\begin{aligned} \texttt{fst}\,(e_1, e_2) &\rightarrow e_1 \\ \texttt{snd}\,(e_1, e_2) &\rightarrow e_2 \end{aligned}$$

Laiendus: summatüüp

- Tüübid: $\tau ::= \ldots \mid \tau_1 + \tau_2$
- Termid: $e ::= \ldots \mid \mathtt{inl} \mid \mathtt{inr} \mid \mathsf{case}\,(e_0;\, x_1.e_1;\, x_2.e_2)$
- Väärtused: $v ::= \ldots \mid \mathtt{inl}\; v_1 \mid \mathtt{inl}\; v_2$
- Tüüpimisreeglid:

$$\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \mathtt{inl}\; e : \tau_1 + \tau_2} \qquad \frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \mathtt{inr}\; e : \tau_1 + \tau_2}$$

$$\frac{\Gamma \vdash e_0 : \tau_1 + \tau_2 \quad \Gamma, x_1 : \tau_1 \vdash e_1 : \sigma \quad \Gamma, x_2 : \tau_2 \vdash e_2 : \sigma}{\Gamma \vdash \mathsf{case}\,(e_0;\, x_1.e_1;\, x_2.e_2) : \sigma}$$

- Evaluation rules:

$$\begin{array}{lcl} \mathsf{case}\,(\mathtt{inl}\; e_0;\, x_1.e_1;\, x_2.e_2) & \to & e_1[x_1 \mapsto e_0] \\ \mathsf{case}\,(\mathtt{inr}\; e_0;\, x_1.e_1;\, x_2.e_2) & \to & e_2[x_2 \mapsto e_0] \end{array}$$