

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:

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

- Tüübikonstruktor $\rightarrow$ on paremassotsiatiivne.

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

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: $\lambda x : \text{Nat}. x + x : \text{Nat} \rightarrow \text{Nat}$.
 - Igal termil on unikaalne tüüp.
- **Curry stiil**: termid on tavalised (s.o. ilma tüüpideta) ja kasutatakse tüübikorrektuse predikaati (tüübituletus).
 - Näide: $\lambda x. x + x : \text{Nat} \rightarrow \text{Nat}$.
 - Termidel on palju tüüpe (näit. $\lambda x. x : \text{Nat} \rightarrow \text{Nat}$, kuid ka $\lambda x. x : \text{Bool} \rightarrow \text{Bool}$).
- Meie kasutame põhiliselt esimest stiili.

Termid ja redutseerimine

- Termide süntaks:

e	$::=$	x	muutuja
	$ $	$e_1 e_2$	aplikatsioon
	$ $	$\lambda x : \tau. e$	abstraktsioon

- 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 $\Gamma = \{x_1 : \tau_1, \dots, x_n : \tau_n\}$.
 - Konteksti Γ doomenit tähistame $\text{dom}(\Gamma) = \{x_1, \dots, x_n\}$.
 - Korrektses tüüpimisrelatsioonis peab olema $\text{FV}(e) \subseteq \text{dom}(\Gamma)$.
- $\Gamma \leq \Delta$ tähendab, et kui $x \in \text{dom}(\Gamma)$, siis $x \in \text{dom}(\Delta)$ 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:

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

$$\frac{\Gamma \vdash e_1 : \sigma \rightarrow \tau \quad \Gamma \vdash e_2 : \sigma}{\Gamma \vdash e_1 e_2 : \tau} \text{ app}$$

kus $x \notin \text{dom}(\Gamma)$.

- Edaspidi tähistame lihtsalt tüübitud λ -arvutust $\lambda \rightarrow$.

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (1):

$$\vdash \lambda x:\tau, y:\sigma. x :$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (1):

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

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (1):

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

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (1):

$$\frac{\frac{\frac{}{\{x:\tau, y:\sigma\} \vdash x : \tau} \text{var}}{\{x:\tau\} \vdash \lambda y:\sigma. x : \sigma \rightarrow \tau} \text{abs}}{\vdash \lambda x:\tau, y:\sigma. x : \tau \rightarrow \sigma \rightarrow \tau} \text{abs}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (2):

$$\begin{array}{c}
 \Gamma \vdash x : \sigma \rightarrow \tau \quad \frac{\Gamma \vdash y : \rho \rightarrow \sigma \quad \Gamma \vdash z : \rho}{\Gamma \vdash yz : \sigma} \\
 \hline
 \Gamma \vdash x(yz) : \tau \\
 \hline
 \Gamma_2 \vdash \lambda z : \rho. x(yz) : \rho \rightarrow \tau \\
 \hline
 \Gamma_1 \vdash \lambda y : \rho \rightarrow \sigma, z : \rho. x(yz) : (\rho \rightarrow \sigma) \rightarrow \rho \rightarrow \tau \\
 \hline
 \vdash \lambda x : \sigma \rightarrow \tau, y : \rho \rightarrow \sigma, z : \rho. x(yz) : (\sigma \rightarrow \tau) \rightarrow (\rho \rightarrow \sigma) \rightarrow \rho \rightarrow \tau
 \end{array}$$

kus

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

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\frac{}{\vdash \lambda x:\tau. x x : }$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\frac{\{x:\tau\} \vdash x x :}{\vdash \lambda x:\tau. x x : \tau \rightarrow} \text{abs}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\frac{\frac{\frac{\overline{\{x:\tau\} \vdash x : \sigma \rightarrow}}{\{x:\tau\} \vdash x x :} \quad \overline{\{x:\tau\} \vdash x : \sigma}}{\vdash \lambda x:\tau. x x : \tau \rightarrow} \quad \begin{matrix} app \\ abs \end{matrix}$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\frac{\frac{\overline{\{x:\tau\} \vdash x : \tau \rightarrow} \quad \overline{\{x:\tau\} \vdash x : \tau}}{\{x:\tau\} \vdash x x :} \begin{array}{l} var \\ app \end{array}}{\vdash \lambda x:\tau. x x : \tau \rightarrow} abs$$

Tüüpimispuu konstrueerimine

Tüüpimispuu konstrueerimise näide (3):

$$\frac{\frac{\overline{\{x:\tau\} \vdash x : \tau \rightarrow \rho} \text{ } var \quad \overline{\{x:\tau\} \vdash x : \tau} \text{ } var}{\overline{\{x:\tau\} \vdash x x : \rho} \text{ } app} \text{ } abs}{\vdash \lambda x:\tau. x x : \tau \rightarrow \rho}$$

Tüüpimispuu konstrueerimine ebaõnnestub!

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.