Parametric Polymorphism

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Pure Type Systems

$\mathcal{T}$	$=$	$\mathcal{C}$	constant
	$\mid$	$\mathcal{V}$	variable
	$\mid$	$\mathcal{T}\,\mathcal{T}$	application
	$\mid$	$\lambda\mathcal{V}{:}\mathcal{T}.\,\mathcal{T}$	abstraction
	$\mid$	$\forall\mathcal{V}{:}\mathcal{T}.\,\mathcal{T}$	dependent function space

Pure Type Systems

$$\text{axiom}\ \frac{}{\vdash c : s}\ c : s \in \mathcal{A}$$

$$\text{start}\ \frac{\Gamma \vdash \mathsf{A} : s}{\Gamma, x : \mathsf{A} \vdash x : \mathsf{A}}$$

$$\text{weakening}\ \frac{\Gamma \vdash \mathsf{A} : \mathsf{B} \qquad \Gamma \vdash \mathsf{C} : s}{\Gamma, x : \mathsf{C} \vdash \mathsf{A} : \mathsf{B}}$$

$$\text{product}\ \frac{\Gamma \vdash \mathsf{A} : s_1 \qquad \Gamma, x : \mathsf{A} \vdash \mathsf{B} : s_2}{\Gamma \vdash (\forall x{:}\mathsf{A}.\ \mathsf{B}) : s_3}\ (s_1, s_2, s_3) \in \mathcal{R}$$

$$\text{application}\ \frac{\Gamma \vdash \mathsf{F} : (\forall x{:}\mathsf{A}.\ \mathsf{B}) \qquad \Gamma \vdash \mathsf{a} : \mathsf{A}}{\Gamma \vdash \mathsf{F}\,\mathsf{a} : \mathsf{B}[x \mapsto \mathsf{a}]}$$

$$\text{abstraction}\ \frac{\Gamma, x : \mathsf{A} \vdash \mathsf{b} : \mathsf{B} \qquad \Gamma \vdash (\forall x{:}\mathsf{A}.\ \mathsf{B}) : s}{\Gamma \vdash (\lambda x{:}\mathsf{A}.\,\mathsf{b}) : (\forall x{:}\mathsf{A}.\ \mathsf{B})}$$

$$\text{conversion}\ \frac{\Gamma \vdash \mathsf{A} : \mathsf{B} \qquad \Gamma \vdash \mathsf{B}' : s \qquad \mathsf{B} =_\beta \mathsf{B}'}{\Gamma \vdash \mathsf{A} : \mathsf{B}'}$$

$S = (\mathcal{S}, \mathcal{A}, \mathcal{R})$, where $\mathcal{S} \subseteq \mathcal{C}$, $\mathcal{A} \subseteq \mathcal{C} \times \mathcal{S}$ and $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$.

λ - cube

$\lambda\!\rightarrow$ is the PTS determined by

$$\lambda\!\rightarrow \quad \boxed{\begin{array}{ll} \mathcal{S} & *, \Box \\ \mathcal{A} & * : \Box \\ \mathcal{R} & (*, *) \end{array}}$$

λ - cube

$\lambda 2$ is the PTS determined by:

$\lambda 2$	$\mathcal{S}$	$*, \square$
	$\mathcal{A}$	$* : \square$
	$\mathcal{R}$	$(*, *), (\square, *)$

λ - cube

System	Set of specific rules			
$\lambda\to$	$(*,*)$			
$\lambda 2$	$(*,*)$	$(\square,*)$		
λP	$(*,*)$		$(*,\square)$	
$\lambda P2$	$(*,*)$	$(\square,*)$	$(*,\square)$	
$\lambda\underline{\omega}$	$(*,*)$			$(\square,\square)$
$\lambda\omega$	$(*,*)$	$(\square,*)$		$(\square,\square)$
$\lambda P\underline{\omega}$	$(*,*)$		$(*,\square)$	$(\square,\square)$
$\lambda P\omega = \lambda C$	$(*,*)$	$(\square,*)$	$(*,\square)$	$(\square,\square)$

CC_ω

CC_ω is a PTS with this specification:

- $\mathcal{S} = \{\star\} \cup \{\Box_i \mid i \in \mathbb{N}\}$
- $\mathcal{A} = \{\star : \Box_0\} \cup \{\Box_i : \Box_{i+1} \mid i \in \mathbb{N}\}$
- $\mathcal{R} = \{\star \rightsquigarrow \star, \star \rightsquigarrow \Box_i, \Box_i \rightsquigarrow \star \mid i \in \mathbb{N}\} \cup \{(\Box_i, \Box_j, \Box_{\max(i,j)}) \mid i, j \in \mathbb{N}\}$

Relaciones

$$\mathsf{A}_1 \rightarrow \mathsf{A}_2 \rightarrow \cdots \rightarrow \mathsf{A}_i \rightarrow s$$

Translación de tipos a relaciones

$[\![\_]\!] : \mathcal{T} \to \mathcal{T}$ (translation from types to relations).

$$
\begin{aligned}
[\![s]\!] &= \lambda \overline{x : s}.\, \overline{x} \to s \\
[\![x]\!] &= x_R \\
[\![\forall x : \mathsf{A}.\ \mathsf{B}]\!] &= \lambda f : \overline{(\forall x : \mathsf{A}.\ \mathsf{B})}.\, \forall \overline{x : \mathsf{A}}.\ \forall x_R : [\![\mathsf{A}]\!]\, \overline{x}.\ [\![\mathsf{B}]\!]\, (\overline{f\, x}) \\
[\![\mathsf{F}\ \mathsf{a}]\!] &= [\![\mathsf{F}]\!]\, \overline{\mathsf{a}}\, [\![\mathsf{a}]\!] \\
[\![\lambda x : \mathsf{A}.\, \mathsf{b}]\!] &= \lambda \overline{x : \mathsf{A}}.\, \lambda x_R : [\![\mathsf{A}]\!]\, \overline{x}.\, [\![\mathsf{b}]\!]
\end{aligned}
$$

Formalizacón de parametricidad

Parametricity. $\vdash \mathsf{A} : \mathsf{B} \implies \vdash [\![\mathsf{A}]\!] : [\![\mathsf{B}]\!]\ \overline{\mathsf{A}}$

System F

types to relations Note that, by definition,

$$[\![\star]\!]\, T_1\, T_2 \;=\; T_1 \to T_2 \to \star$$

function types

$$\begin{aligned}
&[\![\mathsf{A} \to \mathsf{B}]\!] : [\![\star]\!]\,(\mathsf{A} \to \mathsf{B})\,(\mathsf{A} \to \mathsf{B})\\
&[\![\mathsf{A} \to \mathsf{B}]\!]\, f_1\, f_2 \;=\; \forall a_1 : \mathsf{A}.\ \forall a_2 : \mathsf{A}.\\
&\qquad\qquad\qquad [\![\mathsf{A}]\!]\, a_1\, a_2 \to [\![\mathsf{B}]\!]\,(f_1\, a_1)\,(f_2\, a_2)
\end{aligned}$$

That is, functions are related iff they take related arguments into related outputs.

type schemes

$$\begin{aligned}
&[\![\forall A : \star.\ \mathsf{B}]\!] : [\![\star]\!]\,(\forall A : \star.\ \mathsf{B})\,(\forall A : \star.\ \mathsf{B})\\
&[\![\forall A : \star.\ \mathsf{B}]\!]\, g_1\, g_2 \;=\; \forall A_1 : \star.\ \forall A_2 : \star.\ \forall A_R : [\![\star]\!]\, A_1\, A_2.\\
&\qquad\qquad\qquad [\![\mathsf{B}]\!]\,(g_1\, A_1)\,(g_2\, A_2)
\end{aligned}$$

In words, polymorphic values are related iff instances at related types are related.

Referencias I

Barendregt, H. P. (1992).

Lambda calculi with types, volume 2, page 117–309.

Bernardy, J.-P., Jansson, P., and Paterson, R. (2010).

Parametricity and dependent types.

Proceedings of the 15th ACM SIGPLAN international conference on Functional programming - ICFP '10.