TSPL: Polymorphic Lambda Calculus and The Calculus of Constructions

Philip Wadler

Thursday 17 November 2016

1 Polymorphic lambda calculus

The polymorphic lambda calculus, also called System F, was discovered independently by Girard (1972) and Reynolds (1974).

Let A, B, C range over types, and L, M, N range over terms. We write $\Gamma \vdash A : \mathsf{type}$ if A is a well-formed type, and we write $\Gamma \vdash M : A$ if M is a term of type A, where Γ is an environment of pairs of the form $X : \mathsf{type}$ and $x : A$.

$\boxed{\Gamma \vdash_{\mathrm{F}} A : \mathsf{type}}$

$$\text{typ id}\ \frac{(X : \mathsf{type}) \in \Gamma}{\Gamma \vdash X : \mathsf{type}}$$

$$\text{fun}\ \frac{\Gamma \vdash A : \mathsf{type} \qquad \Gamma \vdash B : \mathsf{type}}{\Gamma \vdash (A \to B) : \mathsf{type}} \qquad \text{all}\ \frac{\Gamma,\, X : \mathsf{type} \vdash B : \mathsf{type}}{\Gamma \vdash (\forall X.\, B) : \mathsf{type}}$$

$\boxed{\Gamma \vdash_{\mathrm{F}} M : A}$

$$\text{id}\ \frac{(x : A) \in \Gamma}{\Gamma \vdash x : A}$$

$$\text{fun abs}\ \frac{\Gamma,\, x : A \vdash N : B}{\Gamma \vdash (\lambda x{:}A.\, N) : A \to B} \qquad \text{fun app}\ \frac{\Gamma \vdash L : A \to B \qquad \Gamma \vdash M : A}{\Gamma \vdash (L\ M) : B}$$

$$\text{typ abs}\ \frac{\Gamma,\, X : \mathsf{type} \vdash N : B}{\Gamma \vdash (\Lambda X.\, N) : \forall X.\, B} \qquad \text{typ app}\ \frac{\Gamma \vdash L : \forall X.\, B \qquad \Gamma \vdash A : \mathsf{type}}{\Gamma \vdash (L\ A) : [X \mapsto A]B}$$

The reduction rules are:

$$\begin{aligned} (\lambda x{:}A.\, N)\ M &\longrightarrow [x \mapsto M]N \\ (\Lambda X.\, N)\ A &\longrightarrow [X \mapsto A]N \end{aligned}$$

With the congruence rule:

$$E ::= \Box\, M \mid V\, \Box \mid \Box\, A$$

$$\frac{M \longrightarrow M'}{E[M] \longrightarrow E[M']}$$

Product, unit, sum, and empty types can be defined in terms of these, as can natural numbers.

$$\begin{aligned}
A \times B &\stackrel{\text{def}}{=} \forall Z.\, (A \to B \to Z) \to Z \\
(V, W) &\stackrel{\text{def}}{=} \Lambda Z.\, \lambda k{:}A{\to}B{\to}Z.\, k\, V\, W \\
\mathsf{fst}\ L &\stackrel{\text{def}}{=} L\, A\, (\lambda x{:}A.\, \lambda y{:}B.\, x) \\
\mathsf{snd}\ L &\stackrel{\text{def}}{=} L\, B\, (\lambda x{:}A.\, \lambda y{:}B.\, y)
\end{aligned}$$

$$\begin{aligned}
1 &\stackrel{\text{def}}{=} \forall Z.\, Z \to Z \\
() &\stackrel{\text{def}}{=} \Lambda Z.\, \lambda z : Z.\, z
\end{aligned}$$

$$\begin{aligned}
A + B &\stackrel{\text{def}}{=} \forall Z.\, (A \to Z) \to (B \to Z) \to Z \\
\mathsf{inl}\ V &\stackrel{\text{def}}{=} \Lambda Z.\, \lambda h{:}A{\to}Z.\, \lambda k{:}B{\to}Z.\, h\, V \\
\mathsf{inr}\ W &\stackrel{\text{def}}{=} \Lambda Z.\, \lambda h{:}A{\to}Z.\, \lambda k{:}B{\to}Z.\, k\, W \\
\mathsf{case}\ L\ \mathsf{of}\ \{\, \mathsf{inl}\ x \Rightarrow M;\ \mathsf{inr}\ y \Rightarrow N\,\} : C &\stackrel{\text{def}}{=} L\, C\, (\lambda x{:}A.\, M)\, (\lambda y{:}B.\, N)
\end{aligned}$$

$$\begin{aligned}
0 &\stackrel{\text{def}}{=} \forall Z.\, Z \\
\mathsf{case}\ L\ \mathsf{of}\ \{\,\} : C &\stackrel{\text{def}}{=} L\, C
\end{aligned}$$

$$\begin{aligned}
\mathsf{Nat} &\stackrel{\text{def}}{=} \forall Z.\, (Z \to Z) \to Z \to Z \\
\mathsf{Z} &\stackrel{\text{def}}{=} \Lambda Z.\, \lambda s{:}Z{\to}Z.\, \lambda z{:}Z.\, z \\
\mathsf{S} &\stackrel{\text{def}}{=} \lambda n{:}\mathsf{Nat}.\, \Lambda Z.\, \lambda s{:}Z{\to}Z.\, \lambda z{:}Z.\, s\, (n\, Z\, s\, z) \\
m + n &\stackrel{\text{def}}{=} m\, \mathsf{Nat}\, \mathsf{S}\, n \\
m \times n &\stackrel{\text{def}}{=} m\, \mathsf{Nat}\, (\lambda x{:}\mathsf{Nat}.\, n + x)\, \mathsf{Z} \\
m^n &\stackrel{\text{def}}{=} m\, \mathsf{Nat}\, (\lambda x{:}\mathsf{Nat}.\, n \times x)\, (\mathsf{S}\, \mathsf{Z})
\end{aligned}$$

2 Calculus of Constructions

The calculus of constructions was proposed by Coquand and Huet (1988). It is the basis of the system used in Coq.

Let A, B, C, L, M, N range over constructions (which encompass both terms and types), and let s range over either type or prop, which are called *sorts*. If $\Gamma \vdash A : \mathsf{type}$ then we say A is a type, while if $\Gamma \vdash M : A$ we say term M has type A, where Γ is an environment of pairs of the form $x : A$ (which includes $x : \mathsf{type}$).

Where we previously wrote $\forall X.\, B[X]$ we now write $\forall x{:}\mathsf{type}.\, B[x]$, and where we previously wrote $\Lambda X.\, B[X]$ we now write $\lambda x{:}\mathsf{type}.\, B[x]$.

$\boxed{\Gamma \vdash_{\mathrm{F}} M : A}$

$$\text{id}\,\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \text{type}\,\frac{}{\Gamma \vdash \mathsf{prop} : \mathsf{type}} \qquad \text{all}\,\frac{\Gamma,\, x : A \vdash B : s}{\Gamma \vdash (\forall x : A.\, B) : s}$$

$$\text{abs}\,\frac{\Gamma,\, x : A \vdash N : B}{\Gamma \vdash (\lambda x{:}A.\, N) : \forall x{:}A.\, B} \qquad \text{app}\,\frac{\Gamma \vdash L : \forall x{:}A.\, B \qquad \Gamma \vdash M : A}{\Gamma \vdash (L\ M) : [x \mapsto M]B}$$

$$\text{conv}\,\frac{\Gamma \vdash M : A \qquad \Gamma \vdash B : s \qquad A =_\beta B}{\Gamma \vdash M : B}$$

We have the following abbreviation.

$$A \to B \stackrel{\text{def}}{=} \forall x{:}A.\, B \qquad \text{if } x \notin B$$

Not including congruences, there is only one reduction rule.

$$(\lambda x{:}A.\, N)\ M \longrightarrow [x \mapsto M]N$$

System F is included in the Calculus of Constructions. We can also define many other things, such as equality of terms of type A.

$$(x =_A y) \stackrel{\text{def}}{=} \forall P : A {\to} \mathsf{prop}.\, P\ x \to P\ y$$

References

T. Coquand and G. Huet. The calculus of constructions. *Inf. Comput.*, 76(2-3):95–120, Feb. 1988. ISSN 0890-5401. doi: 10.1016/0890-5401(88)90005-3. URL `http://dx.doi.org/10.1016/0890-5401(88)90005-3`.

J.-Y. Girard. *Interprétation Fonctionnelle et Élimination des Coupures de l'Arithmétique d'Ordre Supérieur.* Thèse de doctorat d'état, Université Paris VII, Paris, France, June 1972.

J. C. Reynolds. Towards a theory of type structure. In *Programming Symposium*, volume 19 of *LNCS*, pages 408–425. Springer-Verlag, 1974.