

Simply Typed λ -calculus

Adrien FRABETTI MATHIEU

Contents

Definition	1
Syntax	1
Semantics	1
Meta-theory	1
Strong normalization	1

Definition

Syntax

For this section, assume we have a (possibly infinite) set $\mathcal{T}$ of *atomic types*. A type τ is defined by the following grammar:

$$\begin{array}{l} \tau \quad := \quad T \in \mathcal{T} \\ \quad \mid \tau \rightarrow \tau \end{array}$$

We equip the λ -calculus with the following type system.

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \text{Ax} \qquad \frac{\Gamma, x : \tau \vdash t : \tau'}{\Gamma \vdash \lambda x. t : \tau \rightarrow \tau'} \qquad \frac{\Gamma \vdash u : \tau \rightarrow \tau' \quad \Gamma \vdash v : \tau}{\Gamma \vdash uv : \tau'}$$

Semantics

Let $\mathcal{C}$ be a cartesian closed category, and a function $\rho : \mathcal{T} \rightarrow \mathcal{C}$ that maps atomic types to objects of $\mathcal{C}$. We can define an interpretation $\llbracket - \rrbracket$ going from types to objects of $\mathcal{C}$, as well as, for any open term t such that $\Gamma \vdash t : \tau$, we have

$$\llbracket t \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket \tau \rrbracket$$

defined by induction on the typing derivation, where

$$\llbracket x_1 : \tau_1, \dots, x_n : \tau_n \rrbracket = \llbracket \tau_1 \rrbracket \times \dots \times \llbracket \tau_n \rrbracket$$

Axiom by definition of a product, there exists a projection $\llbracket \Gamma \rrbracket \rightarrow \llbracket \tau \rrbracket$ if $x : \tau \in \Gamma$

Abstraction by induction hypothesis, we have constructed a morphism $\llbracket t \rrbracket : \llbracket \Gamma \rrbracket \times \llbracket \tau \rrbracket \rightarrow \llbracket \tau' \rrbracket$.

By adjunction, this corresponds exactly to a morphism $\llbracket \lambda x. t \rrbracket = \lambda(\llbracket t \rrbracket) : \llbracket \Gamma \rrbracket \rightarrow \llbracket \tau \rrbracket \Rightarrow \llbracket \tau' \rrbracket$.

Application by induction hypothesis, we have $\llbracket u \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket \tau \rrbracket \Rightarrow \llbracket \tau' \rrbracket$ and $\llbracket v \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket \tau \rrbracket$. By adjunction, the former morphism corresponds exactly to a $\text{app}(\llbracket u \rrbracket) : \llbracket \Gamma \rrbracket \times \llbracket \tau \rrbracket \rightarrow \llbracket \tau' \rrbracket$.

Therefore, we have

$$\llbracket \Gamma \rrbracket \xrightarrow{\Delta} \llbracket \Gamma \rrbracket \times \llbracket \Gamma \rrbracket \xrightarrow{\llbracket \Gamma \rrbracket \times \llbracket v \rrbracket} \llbracket \Gamma \rrbracket \times \llbracket \tau \rrbracket \xrightarrow{\text{app}(\llbracket u \rrbracket)} \llbracket \tau' \rrbracket$$

Meta-theory

Strong normalization

Theorem 1 (Strong normalization)

Soit t un terme (potentiellement ouvert), tel qu'il existe un type τ et un environnement Γ tels que

$$\Gamma \vdash t : \tau$$

t est fortement normalisant.

Theorem (..) $\Rightarrow$... Strong normalization

In order to prove this highly non-trivial result, we need to define an interpretation of types $\llbracket \tau \rrbracket$, by induction on τ ²:

- $\llbracket T \rrbracket := \{u : \text{Term} \mid u \text{ is strongly normalizing}\}$

²Note that, while it might be more intuitive would be to enforce that, if $u \in \llbracket \tau \rrbracket$, then $u : \tau$. This would work, but it turns out this assumption is never used in the rest of the proof, and would add unnecessary complexity.

$$\bullet \llbracket \tau \rightarrow \tau' \rrbracket \quad := \quad \{u : \text{Term} \mid \forall v \in \llbracket \tau \rrbracket, uv \in \llbracket \tau' \rrbracket\}$$