Lambda-calcul simplement typé

1 Introduction

1.1 Introduction au typage

Ω et Y contiennent des termes qui s'appliquent à eux-mêmes.
Le paradoxe du barbier est :

> Le barbier rase tous ceux qui ne se rasent pas eux-mêmes. *Qui rase le barbier ?*

Pour éviter les paradoxes, on cherche à éviter de tels termes.
On va donc typer les termes.
C'est aussi bien pour la programmation.

1.2 Objectifs du typage

Le typage a donc deux objectifs :

- préserver la correction, rien de mauvais ne peut arriver,
- préserver la terminaison, toutes les réductions se terminent.

En λ-calcul la terminaison s'appelle la normalisation forte.

1.3 Notion d' environnement

Il faut typer les variables libres, il faut donc faire des hypothèses sur les types de ces variables.
D'où la notion d'environnement.
Un environnement est un ensemble d'association de types à des variables.

$$\Gamma \equiv x_1 : \sigma_1, ..., x_n : \sigma_n$$

1.4 Les types

Un jugement est l'affirmation du type σ d'un terme M sous un certain environnement Γ :

$$\Gamma \vdash M : \sigma$$

Les types sont

- soit des variables α,
- soit des types applications $\sigma \to \tau$.

1.5 Les règles

$$\text{(Var)} \qquad \Gamma, x:\sigma \vdash x:\sigma$$

$$\text{(Abs)} \qquad \frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash \lambda x:\sigma.M : \sigma \to \tau}$$

$$\text{(App)} \qquad \frac{\Gamma \vdash M:\sigma \to \tau \qquad \Gamma \vdash N:\sigma}{\Gamma \vdash MN:\tau}$$

2 Exercices et théorèmes

2.1 Exercices

Typez les termes

- $B \equiv \lambda xyz.x(yz)$,
- $I \equiv \lambda x.x$,
- $C \equiv \lambda xyz.xzy$,
- $K \equiv \lambda xy.x$,
- $S \equiv \lambda xyz.xz(yz)$.
- $II \equiv (\lambda x.x)(\lambda x.x)$.
- $\mathbf{22} \equiv (\lambda fx.f(fx))(\lambda fx.f(fx))$.

A-t-on le même type pour I (resp. $\mathbf{2}$) dans chaque cas ?

Conclusion : Le système de types simples n'est pas assez général. Il ne permet d'affecter un type unique à chaque terme.

2.2 Théorèmes

$\boxed{\text{Si } \Gamma, x:\sigma \vdash M:\tau \text{ et } \Gamma \vdash N:\sigma \text{ alors } \Gamma \vdash M[x:=N]:\tau}$

Démonstration : Par induction sur M.

3 Réduction du sujet

Lemme Réduction du sujet
La β-réduction préserve le type.

$\boxed{\text{Si } \Gamma \vdash M:\sigma \text{ et si } M \xrightarrow[\beta]{} N \text{ alors } \Gamma \vdash N:\sigma.}$

Démonstration : Par induction sur la définition de $M \xrightarrow[\beta]{} N$.

- $M \equiv (\lambda x:\tau.M_1)M_2$
  $\Gamma \vdash (\lambda x:\tau.M_1)M_2 : \sigma$ vient de $\Gamma, x:\tau \vdash M_1:\sigma$ et de $\Gamma \vdash M_2:\tau$.
  D'autre part $N \equiv M_1[x:=M_2]$
  or d'après le lemme $\Gamma \vdash M_1[x:=M_2]:\sigma$.
- $M \equiv (\lambda x:\tau.M_1)M_2$
  $\Gamma \vdash (\lambda x:\tau.M_1)M_2 : \sigma$ vient de $\Gamma, x:\tau \vdash M_1:\sigma$ et de $\Gamma \vdash M_2:\tau$.
  D'autre part $N \equiv M_1[x:=M_2]$
  or d'après le lemme $\Gamma \vdash M_1[x:=M_2]:\sigma$.

– $M \equiv M_1 M_2$ avec $N \equiv N_1 M_2$ et $M_1 \xrightarrow[\beta]{} N_1$.
$\Gamma \vdash M$ vient de $\Gamma \vdash M_1 : \tau \rightarrow \sigma$ et $\Gamma \vdash M_2 : \tau$,
par induction $\Gamma \vdash N_1 : \tau \rightarrow \sigma$ et donc $\Gamma \vdash N_1 M_2 : \sigma$.

Si $\Gamma \vdash M : \sigma$ et si $M \xrightarrow[\beta]{} N$ alors $\Gamma \vdash N : \sigma$.

Démonstration : Par induction sur la définition de $M \xrightarrow[\beta]{} N$.

– $M \equiv (\lambda x : \tau . M_1) M_2$
$\Gamma \vdash (\lambda x : \tau . M_1) M_2 : \sigma$ vient de $\Gamma, x : \tau \vdash M_1 : \sigma$ et de $\Gamma \vdash M_2 : \tau$.
D'autre part $N \equiv M_1[x := M_2]$
or d'après le lemme $\Gamma \vdash M_1[x := M_2] : \sigma$.
– $M \equiv M_1 M_2$ avec $N \equiv N_1 M_2$ et $M_1 \xrightarrow[\beta]{} N_1$.
$\Gamma \vdash M$ vient de $\Gamma \vdash M_1 : \tau \rightarrow \sigma$ et $\Gamma \vdash M_2 : \tau$,
par induction $\Gamma \vdash N_1 : \tau \rightarrow \sigma$ et donc $\Gamma \vdash N_1 M_2 : \sigma$.
– Les autres cas sont similaires.

3.1 Commentaires

– Si un terme est d'un certain type, il n'en sortira pas par β-réduction.
– Si un terme a une forme normale et s'il a un type, alors sa forme normale a ce type.

4 L'isomorphisme de Curry-Howard

$$
\begin{array}{cc|cc}
(\text{Var}) & \Gamma, x : \sigma \vdash x : \sigma & & \Gamma, p \vdash p \\
(\text{Abs}) & \dfrac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma . M : \sigma \rightarrow \tau} & \Rightarrow I & \dfrac{\Gamma, p \vdash q}{\Gamma \vdash p \Rightarrow q} \\
(\text{App}) & \dfrac{\Gamma \vdash M : \sigma \rightarrow \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} & \Rightarrow E & \dfrac{\Gamma \vdash p \Rightarrow q \qquad \Gamma \vdash p}{\Gamma \vdash q}
\end{array}
$$

Dans $\Gamma \vdash M : \sigma$,
– M : est une annotation qui est le «terme de preuve»,
– σ peut-être vu comme un type ou comme une proposition.

On obtient le tableau de correspondance :

types	propositions
termes	preuves
réduction	simplification des preuves

5 Normalisation forte

Définition :
Un terme M est fortement normalisable si toute suite de réductions $M \xrightarrow[\beta]{} M_1 \xrightarrow[\beta]{} \ldots \xrightarrow[\beta]{} M_n \ldots$ est finie.

Théorème :
Si $\Gamma \vdash \tau$ alors M est fortement normalisable.