

Programmation TP 7 : λ -calcul
 { jrobert, fbouchez, apardon }@ens-lyon.fr
<http://perso.ens-lyon.fr/florent.bouchez/>
 30 octobre 2007

1 Échauffement

Le λ -calcul *fort* est défini par la syntaxe et les quatre règles d'inférence suivantes :

(termes) $M, N ::= x \mid MN \mid \lambda x.M$

$$\frac{}{(\lambda x.M)N \rightarrow M[N/x]} \beta \quad \frac{M \rightarrow M'}{MN \rightarrow M'N} \mu_l \quad \frac{N \rightarrow N'}{MN \rightarrow MN'} \mu_r \quad \left(\frac{M \rightarrow M'}{\lambda x.M \rightarrow \lambda x.M'} \xi \right)$$

La règle ξ est un peu trop puissante : avec l'intuition que $\lambda x.M$ représente une fonction qui attend son argument, elle permet de réduire le corps de la fonction, sans attendre l'argument. Comme ce n'est pas vraiment réaliste dans le cadre des langages de programmation, et on est amenés à retirer cette règle, ce qui nous donne le λ -calcul *faible*.

Même sans cette règle, des choix sont possibles lors de la réduction d'un terme :

Q 1.1 Posons $I = \lambda x.x$ et $F = \lambda x.\lambda y.(y(yI))$. Parmi toutes les séquences de réduction possibles, qui mènent de $F(II)(II)$ à I , donnez-en une de longueur minimale, et une de longueur maximale. Dans le premier cas, explicitez les arbres de dérivation.

On reviendra là-dessus à la section 3.

Q 1.2 Réduisez le terme $\Omega = (\lambda x.xx)(\lambda x.xx)$. Qu'en déduisez-vous ?

Q 1.3 Donnez un terme dont la taille (syntaxique) augmente indéfiniment par réduction.

2 Expressivité

On peut représenter un entier n dans le λ -calcul par une la fonction qui itère n fois une fonction donnée :

$$[n] = \lambda f.\lambda x.\underbrace{f(f(\dots(fx)\dots))}_n$$

Q 2.1 Définissez un encodage de l'addition, $[+]$, tel que $[+][m][n] \rightarrow^* [m+n]$.

Q 2.2 Définissez un encodage pour les couples : $[(M, N)]$, $[fst]$, $[snd]$.

Q 2.3 Soit $A = \lambda xy.y(xxy)$, réduisez le terme $\theta = AA$. A quoi cela peut-il bien servir ?

3 Stratégies

Souvenez vous de la première question : même avec la réduction faible, on a des choix à faire lors de l'exécution d'un terme. Ici, on s'intéresse à définir des stratégies, c'est à dire des jeux de règles qui soient déterministes.

3.1 Appel par nom (cbn)

Une première solution consiste à ne réduire que le redex le plus "à gauche" (s'il existe) : on retire la règle μ_r . C'est la stratégie adoptée dans `Haskell`.

$$\frac{}{(\lambda x.M)N \rightarrow M[N/x]} \beta \qquad \frac{M \rightarrow M'}{MN \rightarrow M'N} \mu_l$$

Q 3.1 Pourquoi les règles précédentes sont-elles déterministes, et réduisent le redex le plus à gauche ?

Q 3.2 Réduisez le terme de la première question selon ces règles. Pourquoi dit-on qu'il s'agit d'appel par *nom* ? Qu'en pensez-vous ?

3.2 Appel par valeur (cbv)

La solution de `ML` consiste au contraire à réduire les arguments avant de les donner à la fonction. Il faut donc définir une notion de *valeur*, c'est à dire de terme complètement réduit :

$$\text{(valeurs)} \quad V ::= x \mid \lambda x.M$$

$$\frac{}{(\lambda x.M)V \rightarrow M[V/x]} \beta \qquad \frac{M \rightarrow M'}{MV \rightarrow M'V} \mu_l \qquad \frac{N \rightarrow N'}{MN \rightarrow MN'} \mu_r$$

(seules les variables et les abstractions sont complètement réduites).

Q 3.3 Réduisez le terme de la première question selon ces règles. Pourquoi dit-on qu'il s'agit d'appel par *valeur* ? Qu'en pensez-vous ?

Q 3.4 Ces règles correspondent à la stratégie implantée dans `OCaml`, mais il y a d'autres stratégies possibles, qui méritent tout autant l'appellation *par valeur*. Définissez une telle stratégie.

Q 3.5 Quel est l'impact des différentes stratégies d'évaluation sur la terminaison ?

4 Encodage et continuations

On peut définir un encodage du λ -calcul dans lui même ($[\cdot] : \Lambda \rightarrow \Lambda$), tel que pour tout terme M , la réduction faible de $[M]$ soit déterministe et corresponde à la réduction de M en appel par valeur (c'est aussi possible en appel par nom). En d'autres termes, l'encodage va respecter le "sens" mais forcer une stratégie d'évaluation. L'idée consiste à insérer des continuations un peu partout dans le terme.

Q 4.1 Définissez un encodage qui réalise l'appel par valeur, "droite-gauche".

Q 4.2 Comment en déduire l'encodage "gauche-droite" ?

Q 4.3 Comment énoncer formellement la propriété de correction "[M] fidèle à M " ?

Q 4.4 Définissez un encodage pour l'appel par nom.