

TD 1 : λ -calcul et codage des types de données

jean-marie.madiot@ens-lyon.org

Exercice 1 *λ -calcul*Le λ -calcul est défini par la syntaxe

$$M, N := x \mid \lambda x. M \mid M N \quad \text{où } x \text{ est une variable}$$

et la règle de réécriture (la β -réduction)

$$(\lambda x. M) N \rightarrow M[N/x].$$

La clôture symétrique réflexive transitive de \rightarrow est notée $=_\beta$. Voici quelques termes classiques :

$$I := \lambda x. x \quad K := \lambda x \lambda y. x \quad S := \lambda x \lambda y \lambda z. x z (y z) \quad \Delta := \lambda x. x x \quad \Omega := \Delta \Delta$$

1. Réduire les λ -termes $\Delta I I$ et Ω .
2. Donner les graphes de réduction des λ -termes $S K K$, $\Delta (I I)$ et $K I \Omega$.

Exercice 2 *Couples et types somme*Définissez des λ -termes pour

$$\begin{array}{ll} \langle -, - \rangle & \text{le constructeur de couples} \\ \pi_1 & \text{la première projection} \\ \pi_2 & \text{la seconde projection} \end{array} \quad \begin{array}{ll} \iota_1 & \text{la première injection} \\ \iota_2 & \text{la seconde injection} \\ \text{case} & \text{le filtrage} \end{array}$$

tels que : $\pi_1 \langle x, y \rangle =_\beta x$, $\pi_2 \langle x, y \rangle =_\beta y$, $\text{case}(\iota_1 x) f g =_\beta f x$, $\text{case}(\iota_2 x) f g =_\beta g x$.**Exercice 3** *Entiers et booléens de Church*La représentation de Church d'un entier n est le λ -terme $\bar{n} := \lambda f x. f^n x$, c'est à dire n itérations de la fonction f en x .

1. Écrire $\bar{0}$ et $\bar{3}$.
2. Écrire une fonction successeur : $S \bar{n} =_\beta \overline{n+1}$.
3. Écrire un itérateur, c'est-à-dire un terme Iter tel que pour tous termes M, N , on ait

$$\text{Iter } M N \bar{0} =_\beta M \quad \text{et} \quad \text{Iter } M N (S \bar{n}) =_\beta N (\text{Iter } M N \bar{n}).$$

4. Écrire des λ -termes représentant l'addition et la multiplication.
5. Quelle fonction représente le terme $\bar{n} \bar{m}$?

On représente les booléens par $\mathbf{T} := \lambda xy. x$ et $\mathbf{F} := \lambda xy. y$.

6. Donner une représentation de **if then else**.

7. Comment représenter les couples ?
8. Proposer un λ -terme représentant le prédécesseur.

Exercice 4 *λ -calcul (bis)*

1. Caractériser les λ -termes en forme β -normale.
2. Restreindre la β -réduction pour implémenter l'appel par nom et l'appel par valeur. Trouver un λ -terme qui distingue ces deux stratégies de réduction.

Exercice 5 *Entiers de Barendregt*

Les entiers de Barendregt $\ulcorner n \urcorner$ sont définis par

$$\ulcorner 0 \urcorner := I \qquad \ulcorner n + 1 \urcorner := \lambda k.k \text{ F } \ulcorner n \urcorner$$

1. Proposer une implémentation du successeur, du prédécesseur et de la conditionnelle.
2. Proposer une implémentation de l'addition.

Exercice 6 *Listes et arbres*

On s'intéresse ici à l'encodage des listes en λ -calcul. On va écrire les listes sous la forme $\lambda c.\lambda n.M[c, n]$. Moralement, une liste est une fonction qui attend un constructeur, une liste vide et qui les manipule. Par exemple, la liste ["Salade"; "Tomate"; "Oignon"] sera représentée par

$$\lambda c.\lambda n.(c \text{ "Salade" } (c \text{ "Tomate" } (c \text{ "Oignon" } n))) .$$

1. Écrire les opérateurs nil et cons.
2. Écrire un *itérateur* fold tel que

$$\text{fold } f \ u \ \text{nil} =_{\beta} u \qquad \text{et} \qquad \text{fold } f \ u \ (\text{cons } a \ l) =_{\beta} f \ a \ (\text{fold } f \ u \ l) .$$

3. Écrire la concaténation et le miroir.
4. Proposer un encodage pour les arbres binaires.