

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

simon.castellan@ens-lyon.fr

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

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

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 $\text{T} := \lambda xy. x$ et $\text{F} := \lambda xy. y$.
6. Donner une représentation de **if then else**.

7. Comment représenter les couples ?
8. En utilisant la représentation des couples vue à l'exercice 2, 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

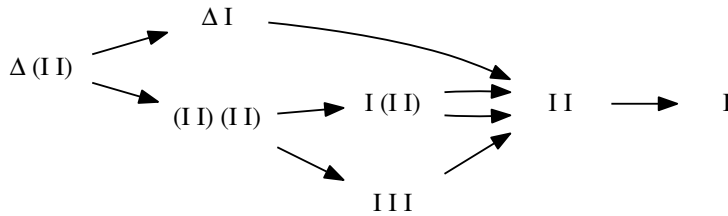
$$\text{fold } f \ u \ \text{nil} =_{\beta} u \qquad \text{et} \qquad \text{fold } f \ u \ (\text{cons } a \ l) =_{\beta} f \ a \ (\text{fold } f \ u \ l) .$$

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

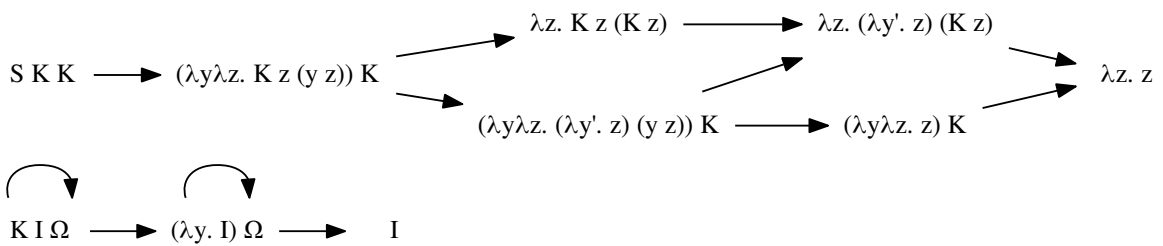
Solutions

► Exercice 1

1. $\underline{\Delta I I} \rightarrow \underline{I I I} \rightarrow \underline{I I} \rightarrow I$
 $\Omega \rightarrow \Omega$



- 2.



► Exercice 2

$$\begin{aligned} \langle -, - \rangle &:= \lambda ab. \lambda f. f a b \\ \pi_1 &:= \lambda p. p T \\ \pi_2 &:= \lambda p. p F \end{aligned}$$

$$\begin{aligned} \iota_1 &:= \lambda x. \lambda f g. f x \\ \iota_2 &:= \lambda y. \lambda f g. g y \\ \text{case} &:= I \end{aligned}$$

► Exercice 3

1. $\bar{0} := \lambda f x. x$ et $\bar{3} := \lambda f x. f(f(f x))$
2. $S := \lambda n f x. \bar{n} f(f x)$ ou $S := \lambda n f x. f(\bar{n} f x)$
3. $\text{Iter} := \lambda M N n. n N M$
 $n \bar{+} m := \lambda f x. \bar{n} f(\bar{m} f x)$
4. $n \bar{*} m := \lambda f x. \bar{n}(\bar{m} f) x$ ou plus simplement $n \bar{*} m := \text{Iter } \bar{n} \text{ plus } \bar{m}$
 $n^{\bar{m}} := \bar{m} \bar{n}$ $n^{\bar{m}} := \text{Iter } \bar{m} \text{ mult } \bar{n}$
5. $\text{if } b \text{ then } M \text{ else } N = b M N$
6. cf exercice 4 : $\langle a, b \rangle := \lambda f. f a b$
7. On utilise le codage des couples. Si on itère $n + 1$ fois la fonction $(x, y) \mapsto (x + 1, x)$, représentée par $\lambda p. \lambda k. k(S(p T))(p T)$ à partir de $(0, 0)$, alors on obtient $(n+1, n)$ D'où $\text{pred } \bar{n} := \bar{n}(\lambda p. \lambda k. k(S(p T))(p T))(\lambda k. k \bar{0} \bar{0})$

► Exercice 4

1. $\vec{\lambda x. x} \vec{N}$ i.e. une suite d'abstraction, puis une variable appliquée à des termes β -réduits.

2. Il suffit de limiter le passage au contexte de la β -réduction. Pour l'appel par valeurs, définissons donc les valeurs :

$$V := x \mid \lambda x. M .$$

Puis on définit la sémantique opérationnelle du λ -calcul en limitant le passage au contexte :

$$\begin{array}{l} \text{call by value} \quad \frac{}{(\lambda x. M) V \rightarrow M[V/x]} \quad \frac{M \rightarrow M'}{M N \rightarrow M' N} \quad \frac{N \rightarrow N'}{V N \rightarrow V N'} \\ \text{call by name} \quad \frac{}{(\lambda x. M) N \rightarrow M[N/x]} \quad \frac{M \rightarrow M'}{M N \rightarrow M' N} \end{array}$$

► **Exercice 5**

- 1.

$$S := \lambda n k. k F n = \lambda n. \langle F, n \rangle \quad P := \lambda n. n F \quad Z := \lambda n. n T$$

- 2.

$$Add := Y(\lambda a x y. Z x y (S (a (P x) y)))$$

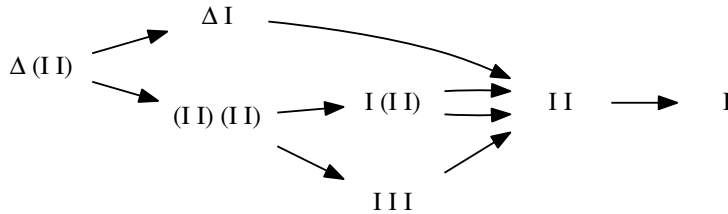
► **Exercice 6**

1. $nil := \lambda c n. n$ et $cons x l := \lambda c n. c x (l c n)$
2. $fold f u l := l f u$
3. $l_1 @ l_2 := \lambda c n. l_1 c (l_2 c n)$ et $rev l := l(\lambda x l'. l' @ (cons x nil)) nil$
4. $leaf := \lambda n l. l$ et $node a t_1 t_2 := \lambda n l. n a t_1 t_2$

Solutions

► **Exercice 1**

1. $\underline{\Delta I I} \rightarrow \underline{I I I} \rightarrow \underline{I I} \rightarrow I$
 $\Omega \rightarrow \Omega$



- 2.

