

# Programmation avancée

## Exercices 5 Lambda-calcul

### 1 Rappels

Un lambda-terme  $t$  est une expression syntaxique qui est soit simplement une variable, soit l'abstraction d'un autre lambda-terme selon une variable, soit l'application d'un terme à un autre :

$$t := x \mid \lambda x.t \mid (t t)$$

où  $x$  est n'importe quel élément d'un ensemble de symboles  $\mathcal{V}$  choisi une fois pour toute comme ensemble de variables. On utilise habituellement les lettres de l'alphabet  $a, b, \dots, x, y, z$  éventuellement indexés par des entiers,  $x_{42}$ , comme ensemble de variables. Il est habituel de raccourcir les notations des lambdas en notant  $\lambda x_1 \dots x_n.t$  au lieu de  $\lambda x_1.\lambda x_2.\dots.\lambda x_n.t$ . Par exemple on écrira  $\lambda xy.x$  au lieu de  $\lambda x.\lambda y.x$ . On doit parfois ajouter des parenthèses autour des sous-expressions  $\lambda x.t$  pour éviter une ambiguïté. Par exemple, on écrit  $((\lambda x.x) y)$ . Par contre on peut omettre nombre de parenthèses d'application en prenant comme convention que les expressions comme  $x y z$  sont parenthésées à gauche :  $((x y) z)$ .

L'ensemble des variables libres d'un lambda-terme est défini par induction sur le terme :

$$\begin{aligned} \text{FV}(x) &= \{x\} \\ \text{FV}(\lambda x.t) &= \text{FV}(t) - \{x\} \\ \text{FV}((t u)) &= \text{FV}(t) \cup \text{FV}(u) \end{aligned}$$

De même, l'ensemble des variables liées d'un lambda-terme est défini par induction sur le terme :

$$\begin{aligned} \text{BV}(x) &= \emptyset \\ \text{BV}(\lambda x.t) &= \text{BV}(t) \cup \{x\} \\ \text{BV}((t u)) &= \text{BV}(t) \cup \text{BV}(u) \end{aligned}$$

Remarque : une variable libre est simplement une variable qui n'est pas liée. Intuitivement, ce sont les lambdas qui lient les variables.

Lorsqu'on écrit un lambda-terme sous forme d'arbre syntaxique, les  $\lambda x$  sont des nœuds unaires, les applications sont des nœuds binaires dénotés usuellement par le symbole @ et les variables correspondent aux feuilles. On peut alors ajouter à cet arbre syntaxique des arcs liant chaque variable

liée au lambda qui la lie. On appelle l'arbre obtenu simplement l'arbre du lambda-terme. Un exemple est donné figure 1.

On vérifie facilement que, partant de la feuille, ce lambda est toujours le premier lambda portant sur le même symbole de variable que l'on rencontre en remontant vers la racine de l'arbre syntaxique. Une variable libre est alors une feuille sans arc sortant.

Deux termes qui ont les mêmes variables libres et dont les arbres ne diffèrent que par le nom des variables liées sont dits  $\alpha$ -équivalents. On peut toujours renommer n'importe quelle variable liée d'un terme ou d'un sous-terme pour obtenir un terme  $\alpha$ -équivalent.

La substitution d'une variable libre d'un terme par un autre terme est une opération qui n'a de sens qu'à  $\alpha$ -renommage près. Si on n'y prend pas garde on risque la capture de variables.

Sur les arbres la substitution d'une variable libre  $x$  par un terme  $u$  dans un terme  $t$  consiste en remplacer dans l'arbre de  $t$  chaque feuille  $x$  non liée par l'arbre de  $u$ . On note  $t[x := u]$  le lambda-terme obtenu.

Un terme de la forme  $((\lambda x.t) u)$  est appelé un rédex. Il se  $\beta$  réduit en une étape en le terme  $t[x := u]$ . Un rédex peut aussi apparaître dans un sous-terme (une sous-expression).

Un terme sans rédex est dit en forme normale.

Un terme peut avoir plusieurs rédex. La  $\beta$ -réduction peut alors s'appliquer à n'importe lequel des sous-termes. Le choix du rédex à réduire peut éventuellement être fixé par une stratégie de réduction.

Partant d'un terme avec rédex si on arrive à une forme normale alors elle est unique, quel que soit le chemin de réduction.

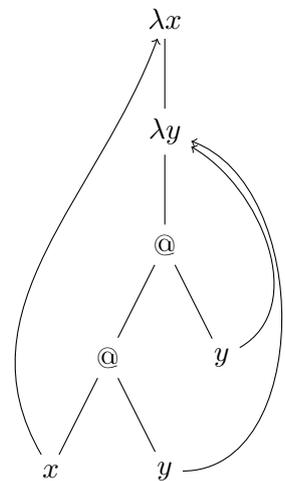


FIGURE 1 – arbre du terme  $\lambda xy.((x y) y)$

## 2 Exercices

### 2.1 Variables libres, variables liées, alpha-équivalence

**Question A. Variables libres et liées.** Donner l'ensemble des variables libres et l'ensemble des variables liées des termes suivants :

- |                         |                                |                                     |
|-------------------------|--------------------------------|-------------------------------------|
| 1. $(x\ y)$             | 3. $\lambda xyz.(x\ (y\ z))$   | 5. $((\lambda x.(x\ y))\ x)$        |
| 2. $(x\ (\lambda y.y))$ | 4. $((\lambda xyz.(x\ y))\ z)$ | 6. $((\lambda x.x)\ (\lambda x.x))$ |

**Question B.  $\alpha$ -équivalence.** Pour chacun des deux derniers termes de l'exercice précédent donner un terme qui lui est  $\alpha$ -équivalent et utilise un maximum de noms de variables.

### 2.2 Substitution, réductions

**Question C. Substitutions.** Effectuer les substitutions suivantes :

- |                               |                               |                                      |
|-------------------------------|-------------------------------|--------------------------------------|
| 1. $x[x := y]$                | 3. $(x\ x)[x := \lambda z.z]$ | 5. $((\lambda x.(x\ y))\ x)[x := z]$ |
| 2. $(x\ y)[x := \lambda z.z]$ | 4. $(x\ x)[y := \lambda z.z]$ | 6. $((\lambda x.(x\ y))\ y)[y := z]$ |

**Question D. Réductions.** Réduire et donner la forme normale des termes suivants :

- |  |  |
|--|--|
| 1. $((\lambda x.x)\ ((\lambda x.x)\ (\lambda x.x)))$ | 4. $((\lambda xy.(x\ (x\ (x\ y))))\ (\lambda x.z))$            |
| 2. $((\lambda x.(x\ x))\ (\lambda z.z))$             | 5. $((\lambda x.(x\ x))\ (\lambda x.(x\ x)))$                  |
| 3. $((\lambda xy.y)\ f\ g)$                          | 6. $((\lambda f.(\lambda x.f(x\ x))\ (\lambda x.f(x\ x)))\ g)$ |

### 2.3 Entiers de Church

On se donne un ensemble de lambda-termes pour représenter les entiers naturels. On représente le zéro par le terme  $\lambda fx.x$ , le un par le terme  $\lambda fx.(f\ x)$ , le deux par le terme  $\lambda fx.(f\ (f\ x))$  et plus généralement l'entier  $n$  est représenté par le terme noté  $\bar{n}$  :

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

Cette représentation est appelée entiers de Church du nom du mathématicien Alonzo Church, qui inventa le lambda-calcul et montra que toutes les fonctions calculables sur les entiers peuvent être représentées par des lambda-termes en utilisant la  $\beta$ -réduction pour effectuer le calcul. Nous allons essayer de calculer un peu avec ses entiers.

**Question E. Successeur.** Donner un lambda-terme  $\overline{succ}$  qui représente l'opération successeur sur les entiers de Church, c'est à dire tel que pour n'importe quel entier naturel  $n$ ,  $(\overline{succ}\ \bar{n})$  se  $\beta$ -réduit en  $\overline{n+1}$ . Tester.

**Question F. Addition.** Donner un lambda-terme  $\overline{add}$  qui représente l'addition de deux entiers.

**Question G. Multiplication.** Même question pour la multiplication  $\overline{mult}$ .