Décomposition d'un lambda-terme

Syntaxe

Un lambda-terme est soit une *application* de la forme ab où a et b sont des lambda-termes, soit une *abstraction* de la forme $\lambda x.t$ où x est une variable et t un lambda-terme.

Les applications sont associatives à gauche dans le sens où abc se décompose comme $(ab)c$. Les abstractions sont associatives à droite dans le sens où $\lambda x.\lambda y.t$ se décompose comme $\lambda x.\ (\lambda y.t)$. Les applications sont prioritaires par rapport aux abstractions dans le sens où $\lambda x.tu$ se décompose comme $\lambda x.\ (tu)$ et pas comme $(\cancel{\lambda x.t})\cancel{u}$.

Par exemple, $\lambda x.\lambda y.abc$ se décompose comme $\lambda x.\ (\lambda y.\ (ab)c)$.

> Donnez la décomposition de $\lambda x.ax\lambda y.bx\lambda z.c$.

Variables libres et liées

Le symbole λ est appelé *lieur* et la variable située immédiatement après est appelée *variable de liaison*. La *portée d'un lieur* est l'abstraction introduite par ce lieur à l'exception des abstractions incluses ayant une variable de liaison de même nom.

Par exemple dans le terme $\lambda x.\ (\lambda y.xy\boxed{(\lambda x.xz)}x)$, la portée du premier lieur est l'ensemble du terme à l'exception de la partie encadrée. Les occurrences de la variable x apparaissant dans cette portée sont dites *liées* entre elles. Les occurrences de x apparaissant dans la zone encadrée sont aussi liées entre elles mais ne sont pas liées aux occurrences de x situées à l'extérieur du cadre. Les occurrences de y sont également liées entre elles. On pourrait renommer le terme de la manière suivante sans changer sons sens : $\lambda x_1.\ (\lambda y.x_1y\boxed{(\lambda x_2.x_2z)}x_1)$.

Les variables qui ne sont pas liées sont dites *libres*. Il n'y en a pas dans l'exemple précédent.

> Identifiez les variables liées entre elles et les variables libres dans le terme suivant :
> $$x\lambda x.\lambda y.xyz$$

Termes équivalents

Il existe des règles de renommage :

1. On peut renommer toutes les occurrences liées entre elles d'une variable à condition qu'aucune variable libre ne soit « capturée », c'est à dire devienne liée.
2. On ne peut pas renommer les variables libres.

Par exemple, dans le terme $\lambda x.\lambda y.xyz$ les deux occurrences de x sont liées entre elles, ainsi que les deux occurrences de y. La variable z est libre. Les occurrences de y peuvent être renommées en t :

$\lambda x.\lambda t.xtz$. Mais elles ne peuvent pas être renommées en z : $\cancel{\lambda x.\lambda z.xzz}$ car alors la variable z située en dernière position, qui est libre, serait capturée.
Si on peut passer d'un terme t à un terme u en respectant les règles de renommage, alors ont dit que t et u sont *équivalents*. La notion est étendue aux termes identiques à l'exception de parenthèse non nécessaires au regard des règles de décompositions vues plus haut (mais qui peuvent améliorer la lisibilité).

Indiquez si les termes suivants sont équivalents :

$$\lambda x.\lambda y.xy(\lambda z.xz)x \underset{?}{\equiv} \lambda x.\lambda y.xy(\lambda x.xx)x$$

$$\lambda x.\lambda y.xy(\lambda z.xz)x \underset{?}{\equiv} \lambda x.\lambda y.xy\lambda z.xzx$$

$$\lambda x.\lambda y.xy(\lambda z.xz)x \underset{?}{\equiv} \lambda x.\lambda y.\ (xy)\ (\lambda z.xz)x$$

$$\lambda x.\lambda y.xy(\lambda z.xz)x \underset{?}{\equiv} \lambda x.\lambda z.xz(\lambda z.xz)x$$

Evaluation d'un terme

Cas simple

On appelle *redex* une abstraction entre parenthèses suivie d'un terme. Par exemple, le terme suivant est un redex :

$$(\lambda t.txt)\lambda y.ty$$

Un redex est caractérisé par une variable de liaison, un corps et un argument. Dans notre exemple, le corps du redex est encadré et son argument est souligné.

$$(\lambda t.\boxed{txt})\underline{\lambda y.ty}$$

Un redex peut être réduit. Cette opération consiste à ne conserver que le corps du redex dans lequel on remplace chaque occurrence de la variable de liaison par l'argument du redex mis entre parenthèses sauf s'il n'y a pas d'ambiguïté. Dans notre exemple, cela donne :

$$(\underline{\lambda y.ty})x(\underline{\lambda y.ty})$$

Ce nouveau terme comporte d'ailleurs un redex.

> Identifiez le redex dans le terme $(\lambda y.ty)x(\lambda y.ty)$ et réduisez-le. Attention de bien décomposer le terme en vous appuyant sur les règles vues dans la fiche précédente.

Cas particulier

Il peut arriver qu'en réalisant une substitution de l'argument du redex, il y ait capture d'une variable libre de l'argument du redex par un lieur situé *dans* le corps du redex. Dans ce cas, il est nécessaire de renommer les occurrences liées entre elles de la variable de liaison concernée. Prenons un exemple. Dans le terme $(\lambda t.\lambda x.tx)x$ le corps du redex est $\lambda x.tx$, sa variable de liaison est t et son argument est une variable libre x. Si on remplace t par x dans le corps du redex, la variable libre x est capturée et on obtient le terme $\lambda x.xx$ qui n'est pas le résultat correct de la réduction.

Il faut alors renommer la variable x du corps du redex, qui devient par exemple $\lambda u.tu$. On peut alors remplacer toutes les occurrences de la variable de liaison du redex, c'est-à-dire t, par l'argument du redex, c'est-à-dire x. On obtient $\lambda u.xu$.

> Chacun des termes suivants contient un redex. Indiquez si un renommage de variable est nécessaire pour le réduire et si oui identifiez la variable concernée. Réduisez ensuite le terme.
> $(\lambda x.xx)\ (\lambda y.xy)$
> $(\lambda x.\lambda y.xy)x\lambda x.x$
> $(\lambda x.\lambda y.xy)\lambda t.yt$

Evaluation d’un terme

Evaluer un terme consiste juste à réaliser des réductions jusqu’à ce qu’il n’y ait plu de redex. S’il y a plusieurs redex, l’ordre dans lequel on les réduira ne changera pas le résultat.

Evaluez le terme $(\lambda x.\lambda y.yx)ab$.